Кристофер Негус Франсуа Каэн

Ubuntu® и Debian® Linux®

для продвинутых

Расширьте свои знания Linux в следующих областях:

- использование консоли
- поиск программ в Сети
- работа с файлами
- прослушивание музыки и просмотр изображений
- обслуживание файловых систем
- резервное копирование данных
- управление запущенными процессами
- управление доступом к сетевым ресурсам
- обслуживание удаленных систем
- настройка безопасности

более 1000 незаменимых команд



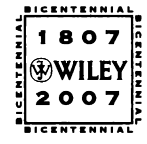



ПИТЕР®

Ubuntu® Linux® TOOLBOX

1000+ Commands for Ubuntu and Debian Power Users

Christopher Negus
François Caen



Wiley Publishing, Inc.

Кристофер Негус   Франсуа Каэн

Ubuntu® и Debian® Linux®

для продвинутых



Москва · Санкт-Петербург · Нижний Новгород · Воронеж
Ростов-на-Дону · Екатеринбург · Самара · Новосибирск
Киев · Харьков · Минск
2011

ББК 32.973.2-018.2
УДК 004.451
К98

Негус К., Каэн Ф.

К98 Ubuntu и Debian Linux для продвинутых: более 1000 незаменимых команд. — СПб.: Питер, 2011. — 352 с.: ил. — (Серия «Для профессионалов»).

ISBN 978-5-94807-027-8

Это руководство научит вас использовать Ubuntu Linux так, как это делают настоящие профессионалы, то есть с помощью командной строки. Вы сможете применять более 1000 команд и получите все необходимое программное обеспечение — начиная с системных утилит, отслеживающих работу вашего ПК и его безопасность, и заканчивая программами для работы в сети и разграничения доступа. Книга дает незаменимые знания и навыки для использования и администрирования настольных ПК и серверов, работающих под управлением Ubuntu, Debian, KNOPPIX и других дистрибутивов Linux.

ББК 32.973.2-018.2
УДК 004.451

ISBN 978-0-470-08293-5 (англ.)
ISBN 978-5-94807-027-8

Краткое содержание

Оглавление

Как обычно, моя работа над книгой посвящается моей жене Шерри.

Кристофер Негус (Christopher Negus)

Моей жене Тоне, которая поддерживает меня во всех моих стараниях.

Франсуа Каэн (François Caen)

Об авторах

Кристофер Негус является автором бестселлеров Fedora and Red Hat Linux Bibles, Linux Toys, Linux Troubleshooting Bible и Linux Bible 2007 Edition. Сейчас он работает в компании Madison Linux Users Group. Прежде чем посвятить себя написанию книг, Крис восемь лет проработал в команде разработчиков операционной системы Linux в компании AT&T, где и была разработана система UNIX. Он также работал в компании Novell над разработкой UNIX и Caldera Linux.

Франсуа Каэн – владелец компании Turbosphere LLC, занимающейся обслуживанием и управлением инфраструктурой бизнес-приложений, которая на 95 % основана на Linux. Будучи сторонником открытых исходных кодов, он читал лекции по администрированию сетей OSS и интернет-сервисам, а также был директором компании Tacoma Linux User Group. Франсуа является сертифицированным инженером Red Hat (RHCE). Свободное время он посвящает управлению компанией Cisco networks.

Благодарности

Я хотел бы поблагодарить Canonical Ltd и команду Ubuntu за их непрерывную и неизменно качественную работу по созданию операционной системы Ubuntu на базе Linux. Особую благодарность хочется выразить Франсуа Каэну, который большую часть прошлого года помогал мне в написании этой книги, закрывая глаза на свои основные профессиональные обязанности. Спасибо Томасу Блейдеру (Thomas Blader), который шагнул гораздо дальше своих основных обязанностей технического редактора, проявив аккуратность и исключительную проницательность при редактировании этой книги. Кроме того, благодарю Эрика Фостер-Джонсона (Eric Foster-Johnson), который присоединился к нам почти в конце работы над книгой, однако существенно помог улучшить описание технических характеристик Ubuntu. Я хотел бы также поблагодарить Дженни Уотсон (Jenny Watson), сотрудницу издательства, за то, что она терпела нас на протяжении всего срока работы над книгой. Наконец, последний, но не менее важный человек, которого я хотел бы поблагодарить, — Сара Шлаер (Sara Shlaer). Спасибо ей за помощь в составлении графиков и бесконечных списков того, что нужно сделать, чтобы эта книга была издана.

Кристофер Негус

Я хотел бы поблагодарить Криса Негуса за то, что он предоставил мне возможность быть соавтором этой книги. Уже несколько лет мы хотели вместе написать книгу, и данное руководство пользователя послужило прекрасным предлогом для совместной работы. Я не смог бы работать над этой книгой без неистощимой поддержки со стороны моей жены Тони. Спасибо ей за то, что мыла посуду все это время, несмотря на то, что мы оба знаем, что это моя работа. Спасибо техническому редактору Томасу Блейдеру за его скрупулезную работу. Спасибо Эрику Фостеру-Джонсону за то, что он привнес в эту книгу свои экспертные знания в области Ubuntu. Спасибо Саре Шлаер за координирование процесса написания книги, а также Дженни Уотсон за проявленное спокойствие. Особую благодарность хочется выразить Уэйну Такеру (Wayne Tucker) за то, что все эти годы он делился со мной своим опытом работы с Debian, в особенности, когда речь шла о последних разработках, таких как kernel 2.4. Наконец, я хотел бы выразить признательность Марку Шаттлворсу (Mark Shuttleworth), компании Canonical Ltd и команде Ubuntu за то, что они делают Ubuntu доступнее и помогают распространять Linux.

Франсуа Каэн

Введение

Огромное Ubuntu-сообщество благодаря своему энтузиазму воспитало многие тысячи новых пользователей Ubuntu Linux. Если вы являетесь одним из них, то, скорее всего, вам очень скоро захочется лучше освоить приложения и графические средства Ubuntu. Вы захотите стать продвинутым пользователем.

Стать продвинутым пользователем любой системы Linux — значит уметь работать с командной строкой: не всякий графический интерфейс может предложить вам те возможности и гибкость, которые доступны при использовании командной строки.

В руководстве пользователя Ubuntu Linux описано более 1000 команд, позволяющих лучше разобраться в системе Linux, и неважно, кем вы являетесь: системным администратором или простым пользователем, — в этой книге рассмотрены команды для создания файловых систем, исправления неполадок в сети, повышения безопасности, а также все, что необходимо знать о Linux.

Эта книга сконцентрирована на изучении команд командной строки Linux (в частности, Ubuntu, дистрибутива Linux, распространяемого через свободное сообщество Ubuntu и спонсируемого компанией Canonical Ltd), а также системы Debian GNU/Linux, на которой и основана Ubuntu. Приобретение навыков, необходимых для управления этими системами, может позволить вам впоследствии работать в собственной системе Linux, а также научиться всему, что необходимо, чтобы быть профессионалом в среде Linux.

buntu берет Linux штурмом

Со времени своего первого издания в 2004 году Ubuntu (www.ubuntu.com) стал самым популярным и, возможно, самым любимым среди пользователей дистрибутивом Linux. Само название дистрибутива, которое переводится как «человечество для других», и то внимание, которое его создатели уделяют поддержке множества других языков и специальных возможностей, отражают основной принцип Ubuntu — принцип бесплатного распространения программного обеспечения за пределами привычного рынка распространения Linux (обычно его формируют приверженцы этой платформы либо люди, которым необходимо обеспечить работу корпоративных серверов).

Разработчики Ubuntu делают все возможное, чтобы операционная система Ubuntu на платформе Linux стала доступнее для неопытных пользователей. Live CD с Ubuntu позволяет ознакомиться с системой перед ее установкой. Если пользователю понравится Ubuntu, то достаточно будет одного щелчка кнопкой мыши, чтобы начать ее установку на жесткий диск; а поскольку дистрибутив Ubuntu создан на базе Debian GNU/Linux, он может использовать огромное количество

программ из репозиториев Debian, являющихся абсолютно бесплатными для пользователей Ubuntu.

То, что разработчики Ubuntu уделяют большое внимание разработке простой в использовании графической оболочки, не значит, что Ubuntu не обладает коммерческой значимостью Linux. Компания Canonical Ltd предлагает разработчикам финансирование через свои службы всемирной поддержки (www.ubuntu.com/support/paid), а также учебные курсы, чтобы помочь вам стать тренинг-партнером Ubuntu (Ubuntu Training Partner) (www.ubuntu.com/support/training). Другими словами, для тех, кто учится работать с Ubuntu, существуют перспективы профессионального роста.

Для кого предназначена книга

Эта книга предназначена для всех, кто хочет в качестве системного администратора или простого пользователя получить доступ к неограниченным возможностям, предоставляемым Linux. Вы можете быть приверженцем платформы Linux, специалистом по Linux или просто профессиональным пользователем, который все чаще отдает себе отчет в том, что Windows вытесняется системами на платформе Linux.

Основным критерием является то, что вы хотите найти быстрые и эффективные способы максимально продуктивной работы в Ubuntu и других системах на базе Debian. Эти системы могут функционировать на нескольких настольных компьютерах на работе, в качестве файл-сервера или принт-сервера в школе, в качестве веб-сервера дома или использоваться просто для развлечения.

Желательно, чтобы вы уже имели опыт работы с Linux. Однако если вы профессиональный пользователь с навыками работы в других системах, например Windows, вам не составит труда освоить использование описываемых в этой книге специфических команд.

Какие темы раскрываются в издании

Эта книга не является пособием для новичков в сфере Linux. Прежде чем серьезно начать изучение данного руководства, необходимо получить общие представления о том, что такое Linux, как работает командная оболочка (shell), а также что представляют собой процессы, файловые системы и сетевые интерфейсы. Впоследствии эта книга дополнит ваши знания информацией, с помощью которой вы сможете выполнять следующие действия.

- **Устанавливать программное обеспечение** — Ubuntu предоставляет возможность использовать для добавления и удаления программ утилиту GUI, с помощью которой можно получать новое программное обеспечение. Вы научитесь с помощью таких инструментов, как `apt-get`, искать, скачивать, устанавливать, обновлять и управлять программным обеспечением из командной строки.
- **Использовать командную оболочку** — в книге доступно описаны способы и приведены подробные рекомендации по использованию командной оболочки.

- **Работать с мультимедиа** — вы научитесь проигрывать и осуществлять потоковую передачу (стримминг) мультимедийного содержимого. Вы также сможете изменять аудиофайлы и изображения, а затем конвертировать их в разные форматы.
- **Работать с файлами** — вы сможете использовать различные типы файлов в Linux, управлять ими, конвертировать и обеспечивать их защиту.
- **Администрировать файловую систему** — вы научитесь управлять правами доступа, размечать и создавать разделы, а также контролировать состояние запоминающих устройств (жестких дисков, CD/DVD-приводов, дискет, USB-накопителей и т. д.), а затем создавать, размечать и проверять файловые системы на этих устройствах.
- **Выполнять резервное копирование и восстановление данных** — вы сможете использовать простые команды для группировки, архивации и упаковки файлов в резервные архивы, тем самым обеспечивая эффективное резервное копирование данных. Затем вы сможете сохранить эти архивы на своем или удаленных компьютерах.
- **Работать с процессами** — данные о текущих процессах могут отображаться исходя из загрузки центрального процессора, использования ресурсов процессора или в виде идентификатора процессов. Вы можете изменять приоритет текущих процессов, делая их фоновыми или приоритетными. Кроме того, существует возможность отправлять процессам команды, в соответствии с которыми эти процессы должны заново прочитывать конфигурационные файлы, останавливать и возобновлять процессы или полностью их завершать (отменять).
- **Управлять системой** — вы сможете вводить команды для проверки системных ресурсов, например использования оперативной памяти, уровня выполнения задачи программой, загрузчиков и модулей ядра.
- **Контролировать сети** — вы научитесь создавать и управлять проводными и беспроводными сетями, а также соединениями с использованием телефонной линии через модем; проверять маршрутизацию, DNS (Domain Name Service) и информацию о хост-машине, следить за сетевым трафиком.
- **Получать доступ к сетевым ресурсам** — прочитав книгу, вы сможете подключаться к удаленным файловым системам Linux и Windows с помощью FTP, NFS и административных средств Samba, а также использовать консольные команды для работы в Интернете.
- **Осуществлять удаленное администрирование** — вы сможете получать доступ и администрировать другие компьютеры, используя удаленный доступ (ssh, telnet и т. д.) и удаленный экран, узнаете об интерфейсах удаленного администрирования, таких как Webmin, SWAT и CUPS.
- **Ограничивать доступ в целях безопасности** — вы научитесь устанавливать межсетевые экраны доступа и пароли, чтобы обезопасить свою систему.
- **Получать справочные сведения** — приложения в конце книги содержат более полную информацию о командной оболочке (например, касательно метасимволов и переменных командного процессора), а также о состоянии системы (используя файловую систему `/proc`).

Будем надеяться, что если мы все сделали правильно, то для поиска нужных команд или GUI-приложений эту книгу будет использовать удобнее, чем Интернет. После того как вы освоите большую часть того, что описано в книге, вы получите в свое распоряжение следующее.

- **Сотни команд** — поскольку вся эта информация сгруппирована в одной небольшой книге, которую легко можно взять с собой, вы получите быстрый доступ к сотням полезных команд, используемых в составе более чем 1000 командных строк.
- **Важную информацию по Linux** — в этой книге приводятся ссылки на интернет-ресурсы, которые помогут вам преуспеть в изучении Linux вообще и Ubuntu в частности.
- **Взаимозаменяемые знания** — большинство команд и параметров, используемых в Ubuntu, также работают и в других системах Linux, несмотря на то что в различных дистрибутивах Linux можно столкнуться с разными графическими оболочками и даже в рамках одного дистрибутива графические средства модифицируются чаще, чем команды.
- **Быстрое решение проблем** — за время, за которое другие успеют только включить компьютер и запустить графическую оболочку, вы уже сможете ввести полдюжины команд и решите проблему.
- **Качество, испытанное временем,** — многие команды, описанные в книге, использовались и в более ранних операционных системах UNIX, поэтому вы получите в свое распоряжение средства, отражающие опыт, накопленный экспертами более чем за 30 лет.

Поскольку вся документация, касающаяся команд, используемых в Linux, занимает тысячи MAN-страниц, текстовой информации и справочных сообщений, вам наверняка придется периодически искать информацию вне этой книги. К счастью, Ubuntu и другие системы на платформе Linux изначально включают в себя достаточно полезной информации. В гл. 1 рассказывается, как получить доступ к информации, которая, возможно, уже присутствует в вашей системе.

Как построена книга

Эта книга не является на 100 % ни справочником (в котором элементы, как правило, перечислены в алфавитном порядке), ни учебным пособием (в котором обычно приводятся пошаговые рекомендации по выполнению тех или иных задач). Вместо этого книга разделена по темам, и мы постарались включить в нее столько полезных команд и параметров, сколько было возможно.

В самом начале гл. 1 дается общая информация о том, что представляет собой Ubuntu и как эта система взаимодействует с другими системами на базе Linux, такими как, например, дистрибутивы Debian. Затем в ней описываются некоторые ресурсы, которые призваны помочь укрепить приобретенные с помощью этой книги знания (например, MAN-страницы, пояснительные тексты и инструкции). В гл. 2 осуществлен краткий обзор процесса установки, а затем описаны полезные

команды, такие как, например, apt-get, позволяющие загружать программное обеспечение для Ubuntu и управлять им.

В гл. 3–6 описаны команды, которые могут показаться полезными постоянному пользователю Linux. В гл. 3 рассмотрены способы использования командной оболочки, в гл. 4 — команды, предназначенные для работы с файлами, в гл. 5 — операции по работе с текстом, а в гл. 6 рассказывается, как работать с изображениями и музыкой.

Начиная с гл. 7 приводятся разделы, посвященные администрированию системы. Вопросы по созданию и проверке файловых систем раскрываются в гл. 7, в то время как команды, используемые для создания резервных копий данных, описываются в гл. 8. Глава 9 посвящена управлению текущими процессами, а в гл. 10 описывается работа приложений, с помощью которых можно управлять основными компонентами и параметрами, такими как аппаратные модули, использование процессора и оперативной памяти.

С гл. 11 начинается обучение управлению сетевыми ресурсами. В ней рассказывается, как установить и использовать проводную и беспроводную сети, а также модемное подключение. В гл. 12 описываются текстовые команды, предназначенные для поиска и просмотра информации в Интернете, передачи файлов и предоставления их для общего пользования, использования служб передачи текстовых сообщений и электронной почты. Средства осуществления удаленного системного администрирования описаны в гл. 13.

В гл. 14 речь идет о средствах ограничения доступа в целях обеспечения безопасности, таких как защитные экраны и ведение журналов. В самом конце книги есть три приложения, содержащие справочные сведения, касающиеся редактирования текста, компонентов центрального процессора (метасимволы и переменные), а также сведения о настройках системы (из файловой системы /proc).

Что нужно для использования данной книги

Хоть мы и надеемся, что вам понравится наше красноречие, это не значит, что вы должны устроиться поудобнее с этой книгой и бокалом вина у камина. Мы надеемся, что вы сядете напротив монитора компьютера и попытаетесь подключиться к сети, отладить файловую систему или добавить пользователя. Вино — по желанию.

Другими словами, предназначение этой книги — быть вашим товарищем во время изучения операционных систем Ubuntu или Debian. Если ни одна из этих систем еще не установлена у вас на компьютере, обратитесь за необходимой информацией по их установке к гл. 2.

Все команды, описанные в этой книге, были протестированы на Ubuntu на архитектурах x86 и x86_64. Однако, поскольку многие из этих команд активно использовались длительное время (некоторые из них уже на 30 лет старше современных приложений UNIX), большинство из них будет работать на системах Debian именно так, как описано в этой книге, независимо от архитектуры процессора.

Кроме того, многие из описанных в этой книге команд будут точно так же работать и на других операционных системах Linux и UNIX. Поскольку данная книга сосредоточена именно на Ubuntu, наиболее заметно отличаться от других операционных систем Linux будут только описания создания пакетов программ, установки и использования средств управления GUI.

Условные обозначения

Чтобы помочь вам почерпнуть как можно больше информации из текста и успевать отслеживать выполняемые операции, на протяжении всей книги мы использовали некоторое количество сокращений. В частности, мы создали свой способ написания команд, который позволил нам включить в эту книгу максимальное количество командных строк.

В примерах команд данные, выводимые компьютером (указания и сообщения, выводимые на командную строку), напечатаны обычным шрифтом, данные, вводимые в компьютер (текст, набираемый пользователем), напечатаны полужирным шрифтом, а краткие пояснения (если они есть) — курсивом. Например:

```
$ ls *jpg        Перечислить все JPEG-файлы в данной папке
hat.jpg
dog.jpg
...
```

Данные, выводимые компьютером, для уменьшения объема иногда даются в сокращенном виде (или полностью опускаются). Содержащиеся в текстах команд многоточия (...) означают, что дополнительные выводимые данные были опущены. Если команда слишком длинная, то в конце каждой строки ставится обратный слэш, обозначающий, что ввод данных продолжается до следующей строки, например:

```
# oggenc NewSong.wav -o NewSong.ogg \
 -a Bernstein -G Classical     \
 -d 06/15/1972 -t "Simple Song"\
 -l "Bernsteins Mass"          \
 -c info="From Kennedy Center"
```

Подобная запись обозначает одну команду. Для получения всей содержащейся в приведенном примере информации в виде одной строки просто введите ее, не используя слэши.

Хотя обычный пользователь в Ubuntu может использовать множество команд, для выполнения некоторых из них необходимо обладать root-привилегиями. Поскольку Ubuntu устанавливается без пароля root, для выполнения административных команд допускается использование команды sudo, не требующей завершения текущего сеанса пользователя, например:

```
chris@host1:/tmp$ sudo useradd -m joe
```

Для большей ясности и с целью экономии места в качестве маркера начала строки обычного пользователя в книге используется значок доллара ($):

```
$ Обозначает командную строку обычного пользователя
```

Время от времени в книге будет встречаться и символ в виде решетки (#), указывающий, что команда должна быть выполнена с root-привилегиями. Таким образом, если вы увидите символ #, то должны будете либо ввести в начале командной строки команду sudo, либо получить root-привилегии с помощью одного из способов, описанных в гл. 3.

Замечания и предупреждения выглядят следующим образом:

ПРИМЕЧАНИЕ

Предупреждения, замечания и советы будут оформлены так же, как и здесь.

ВНИМАНИЕ

Такая врезка привлекает внимание к главной информации, которая не должна быть пропущена.

Что касается внешнего вида текста, то:

- впервые встречающиеся важные термины и слова в книге выделены *курсивом*;
- клавиатурные сокращения мы показываем следующим образом: Ctrl+A (если в соответствии с командой нужно будет ввести прописную букву, сочетание будет выглядеть следующим образом: Ctrl+Shift+A);
- имена файлов, названия команд выделены `моноширинным шрифтом`;
- названия папок, меню, элементов интерфейса, URL-адреса выделены **специальным шрифтом для названий**;
- наконец, для акцентирования внимания на следующей далее команде используется еще один способ выделения текста. Например, может быть написано что-то вроде «используйте следующую команду **для отображения содержимого файла**». В данном случае выделение используется для того, чтобы дать читателям быстрые наглядные подсказки. Таким образом, вы сможете быстро найти нужную команду на странице, если точно знаете, что она находится именно на этой странице.

От издательства

Ваши замечания, предложения и вопросы отправляйте по адресу электронной почты gromakovski@minsk.piter.com (издательство «Питер», компьютерная редакция).

Мы будем рады узнать ваше мнение!

На сайте издательства http://www.piter.com вы найдете подробную информацию о наших книгах.

1 Знакомство с Ubuntu Linux

Вы ежедневно используете Ubuntu Linux на работе или пользуетесь этой системой лишь изредка — неважно: книга, описывающая эффективные и понятные пути поддержания, администрирования, обеспечения защиты и расширения возможностей Ubuntu, может быть бесценным ресурсом, которым как раз и является данная книга.

Эта книга ориентирована в первую очередь на продвинутых пользователей и системных администраторов. Чтобы снабдить вас всеми необходимыми знаниями, мы расскажем вам, как быстро найти и установить программное обеспечение для Ubuntu, а также как обновлять систему, следить за ее состоянием и поддерживать его. Иными словами, мы покажем вам наиболее эффективные способы работы в Ubuntu, основанные на использовании самых мощных инструментов — ваших пальцев.

Основной нашей целью было сгруппировать в одной книге, которую можно легко взять с собой, максимально возможный объем информации. В связи с этим в книге описаны:

- **команды** — в книге содержится большое количество примеров командных строк, демонстрирующих удобные способы управления командной строкой, которая зачастую кажется неукротимой;
- **приложения GUI** — здесь приведены советы по использованию приложений графического интерфейса пользователя для администрирования и настройки Ubuntu;
- **репозитории с программным обеспечением** — кроме того, в книге вы найдете описания методов загрузки и установки программного обеспечения, специально разработанного для Ubuntu;
- **онлайн-ресурсы** — здесь рассказано, где можно найти полезную информацию по Ubuntu: рассылки по электронной почте, на которые можно подписаться, IRC-каналы и другие онлайн-ресурсы;
- **локальная документация** — наконец, в книге рассмотрены приложения для работы с MAN-страницами, стандартными справочниками Linux и UNIX, содержащими специальную документацию по устанавливаемому вами программному обеспечению.

Поскольку эта книга предназначена для опытных пользователей Linux, в ней практически отсутствуют скриншоты значков и меню. Взамен этого вы получите

возможность максимально быстро научиться использовать богатые возможности Ubuntu.

Знания, которые вы почерпнете из этой книги, помогут вам стать более опытным пользователем операционных систем Ubuntu, Debian, а также Linux. Если это кажется вам заманчивым, тогда начнем.

Ubuntu, Debian и Linux

Ubuntu – это операционная система на базе Debian GNU/Linux (www.debian.org). Система Debian была очень популярна в начале 1990-х годов и, будучи полностью самодостаточной, считалась лидирующим дистрибутивом Linux с точки зрения стабильности и безопасности. Debian также известна верностью принципам распространения бесплатного программного обеспечения (www.debian.org/intro/free). Система Ubuntu формировалась именно на ее основе.

Если вы представите себе, что Linux – это что-то вроде торта в кондитерской, то ядро и файлы операционной системы будут не чем иным, как мягким вязким тестом, используемым для приготовления этого торта. Может, это сравнение не дает полного представления, но зато оно наглядно. В случае с дистрибутивами Linux все наоборот: с их помощью делают окантовку «торта» сверху и снизу, раскрашивают его, выкладывают слоями, выравнивают по краям, покрывают глазурью, добавляют ароматические добавки, украшают, ставят свечи, посыпают орехами и присыпкой. Ubuntu, оформленная Canonical Group (www.canonical.com), является глазурью и красителем дистрибутива Linux, построенного на самой вершине торта Debian.

Система Debian дала начало не только Ubuntu, но и многим другим дистрибутивам Linux (www.debian.org/misc/children-distros):

- Xubuntu – персональная операционная система на базе Ubuntu, основанная на Xfce;
- Kubuntu – персональная операционная система на базе Ubuntu, основанная на KDE;
- Edubuntu – производная от Ubuntu операционная система для учебных учреждений;
- Knoppix – Live CD с персональной операционной системой на базе Debian, основанный на KDE;
- Kanotix – Live CD на базе Debian;
- Damn Small Linux – миниатюрный (50 Мбайт) Live CD на базе Knoppix;
- Mepis – Live CD с персональной операционной системой на основе Ubuntu и Debian.

Xubuntu, Kubuntu и Edubuntu являются, по сути, одинаковыми дистрибутивами Ubuntu на базе Debian. Единственное их отличие состоит в используемой по умолчанию среде **Рабочего стола** и пакетах поставляемого с ними программного обеспечения. Например, система Kubuntu укомплектована графической оболочкой KDE и менеджером программных пакетов Adept, которыми не укомплектована

Ubuntu. Edubuntu же предназначена для работы с учебными приложениями, многие из которых не установлены по умолчанию на другие дистрибутивы Ubuntu.

Поскольку Debian и Ubuntu являются открытыми операционными системами, многие составные части которых созданы проектом GNU на основе общедоступной лицензии (General Public License, www.gnu.org/copyleft/gpl.html), любой желающий может получить исходный текст или любую часть системы GPL и изменять, разбирать по частям, достраивать, расширять, внедрять, переделывать и впоследствии бесплатно распространять эти изменения или модификации. Единственным условием является то, что вы должны соблюдать требования, предписываемые GPL, согласно которым любые изменения, произведенные в существующей продукции GPL, должны быть доступны другим для равного использования (для ознакомления с другими лицензиями, признаваемыми Debian, посетите сайт www.debian.org/social_contract).

Таким образом, вы получаете не просто систему с бесплатной онлайн-базой технической поддержки, доступной во всем мире, но и постоянно развивающийся продукт, силы в который вкладывают люди, которым действительно нравится то, что они делают. Многие другие дистрибутивы Linux имеют те же преимущества, однако, если говорить о популярности, Ubuntu вырвалась вперед.

ПРИМЕЧАНИЕ

Никогда не задумывались, откуда у Ubuntu (Edgy Eft) и Debian (Woody) такие странные названия? Ответ вы можете найти на странице https://wiki.ubuntu.com/DevelopmentCodeNames или www.debian.org/doc/manuals/project-history/ch-releases.en.html.

Ubuntu в сравнении с другими дистрибутивами Linux

Если вы откроете командную строку Ubuntu, Linux или Fedora от Red Hat Enterprise, то увидите, что между ними очень мало различий. У обеих систем одинаковые папки и приложения, по функциональности они абсолютно идентичны. Так что же тогда отличает Ubuntu от других дистрибутивов Linux? Единственное отличие состоит в программе установки.

Весь запуск и установка Ubuntu сведены к нескольким щелчкам кнопкой мыши, поскольку многие этапы установки были автоматизированы на основании требований и желаний среднестатистического пользователя. Система Red Hat же, наоборот, предоставляет пользователю массу настроек установки: например, можно выбирать между рабочей станцией и сервером, отдельно выбирать устанавливаемые пакеты и настраивать параметры администрирования.

Другим важным критерием отличия дистрибутивов Linux друг от друга являются приложения по управлению программным обеспечением. Цель приложений и систем по управлению программными пакетами является общей для Debian и других дистрибутивов Linux, однако существенно отличаются действия с ними и их реализация. Ubuntu и большинство других дистрибутивов на базе Debian для управления программным обеспечением используют приложения группы APT (Advanced Package Tool), которые могут быть использованы для установки, уда-

ления, поиска и обновления пакетов Debian (DEB). В Red Hat для выполнения тех же задач с программными пакетами RPM используется система RPM.

Другим существенным отличием является внешний вид систем: их вид при загрузке, экран входа в систему, графическая оболочка, стоящая по умолчанию, «обои» для **Рабочего стола**, набор значков и т. д. Есть довольно много различий по этим внешним признакам. Хотя и Red Hat, и Ubuntu в качестве администратора многооконного режима используют среду **Рабочего стола** GNOME, приложения GUI для администрирования системы и их расположение в раскрывающихся меню полностью различаются.

К тому же экран входа в систему Ubuntu и предлагаемая по умолчанию тема, оформленная в осенних тонах, довольно сильно отличают этот дистрибутив от других. Если вы раскроете меню на **Рабочем столе** Ubuntu, то увидите не длинный список приложений и служб, а простой и изящный набор функциональных программ из всех доступных на **Рабочем столе** Linux. Это сделано, чтобы избавить пользователя от ощущения загромождения, и этот подход хорошо характеризует Ubuntu.

Другой уникальной характеристикой Ubuntu является блокирование учетной записи суперпользователя, что требует использования команды sudo (**www.gratisoft.us/sudo/intro.html**), позволяющей выполнять команды на правах суперпользователя с целью осуществления задач по системному администрированию (в гл. 3 вы найдете подробное описание команд sudo). Вход в систему Linux на правах ее администратора дает возможность получать неограниченный доступ практически ко всем компонентам системы — имея права суперпользователя, было бы слишком просто удалить целую файловую систему, поэтому в Ubuntu использование этой учетной записи ограничено и возможно только для разумных целей. В большинстве дистрибутивов Linux от пользователя требуется войти в систему под собственным именем или использовать команду su, чтобы иметь возможность выполнять административные задачи, в то время как пользователь Ubuntu осуществляет это посредством команды sudo, используя свой собственный пароль входа в систему, а не отдельный, предназначенный для суперпользователя.

Ubuntu обладает уникальными свойствами, хотя и имеет свои недостатки и преимущества. Ubuntu содержит подходящие средства, позволяющие индивидуально настраивать систему, модифицировать ее, экспериментировать и вдаваться во все сложности системы сколько душе угодно, если это на самом деле необходимо. Иными словами, основная идея состоит в том, что это должна быть простая в обращении, безопасная система с понятным и кратким набором прикладных программ, которая не должна ни ограничивать, ни быть перегруженной. Все это делает Ubuntu весьма гибкой системой, работу с которой можно начать без глубокого ее изучения и очень скоро стать специалистом по ее использованию.

Ссылки на ресурсы Ubuntu

Сообщество Ubuntu обладает обширным комплексом знаний, которые доступны в форме онлайн-ресурсов. Ниже приведены ссылки на некоторые наиболее популярные и полезные сайты.

- http://ubuntuforums.org — на этом форуме со встроенными параметрами поиска и модерируемой социальной сетью обитает разноплановое, талантливое и модерируемое сообщество пользователей Ubuntu, а также служба поддержки. Здесь люди делятся своими успехами и неудачами, предоставляют друг другу помощь и поддержку. Если у вас возникнут какие-либо трудности с освоением Ubuntu, то вполне возможно, что кто-то на этом форуме уже сталкивался с теми же проблемами и нашел их решение.
- http://www.ubuntu.com/support — этот сайт предоставляет платные услуги компании Canonical Ltd, которая и создала Ubuntu. Если вы не хотите тратить время на поиск информации на форумах или ожидать ответов на свои вопросы, то можете обратиться в данную компанию по телефону, задать вопрос по электронной почте или на сайте компании. Стоимость этих услуг составляет $20 в месяц. На этом сайте также доступны тренинги, предназначенные для компаний и корпоративных пользователей.
- https://help.ubuntu.com — этот сайт содержит самую свежую официальную онлайн-документацию по каждой версии Ubuntu.
- http://screencasts.ubuntu.com — здесь можно найти обучающие видеоролики, призванные помочь вам научиться выполнять различные команды в Ubuntu: начиная с установки принтера и заканчивая настройкой совместного доступа к файлам через Samba и установкой обновлений для поддержания системы в надлежащей форме. На этом сайте пользователям Ubuntu также предлагается сотрудничество с командой Ubuntu Screencasts Launchpad (https://launchpad.net/~ubuntu-screencasts).
- https://lists.ubuntu.com/mailman/listinfo/ubuntu-users — здесь вы можете подписаться на рассылки по электронной почте для пользователей Ubuntu и обсуждать и решать любые проблемы, начиная от касающихся вопросов управления базами данных mysql и заканчивая установкой сложных сетевых устройств. С архивом обсуждений можно ознакомиться по адресу https://lists.ubuntu.com/archives/ubuntu-users.
- https://wiki.ubuntu.com/IRCResourcePage — если вам нужна поддержка IRC-чата, то можете посетить сайт компании Ubuntu IRC, на котором содержатся руководства, программы-клиенты и чат-серверы. Все они являются бесплатными и доступными в любое время источниками информации. Если вы еще не принимали участия в IRC-чатах, советуем посетить страницу, посвященную правилам поведения в них (www.ubuntulinux.org/community/conduct/).

Если вы планируете приобрести аппаратные средства для использования с Ubuntu или другой системой на базе Linux, то следующие сайты могут помочь определиться с покупкой.

- http://www.linux-usb.org — целью данного сайта является поддержание практических знаний об USB-устройствах, совместимых с Linux. Здесь предусмотрена поисковая система, куда вы можете ввести название или модель интересующего вас устройства и сразу же получить отчет о его совместимости с Linux.
- http://www.linuxfoundation.org/en/OpenPrinting — система печати CUPS (http://cups.org) является стандартной и используется сегодня на большинстве систем

на базе Linux, однако если модель вашего принтера не отображается при попытке добавления его в систему, то попробуйте скачать с данного сайта обновленный PPD-файл и добавить его в свою систему CUPS. На сайте www.linuxfoundation.org/en/OpenPrinting/Database/SuggestedPrinters можно также ознакомиться со списком всех производителей поддерживаемых устройств.

- http://www.sane-project.org — этот сайт посвящен решению вопросов сканирования документов в Linux. Если вы планируете приобретение сканера или многофункционального устройства, посетите этот сайт и узнайте, какие производители уделяют внимание вопросу совместимости с Linux.
- http://tldp.org — данный сайт принадлежит проекту, занимающемуся сбором информации о Linux. Здесь представлен самый большой архив руководств, статей, FAQ, отвечающих на все вопросы, начиная с того, как с помощью Linux сварить чашку кофе, и заканчивая тем, как настроить QoS (quality of service — качество обслуживания) и контролировать трафик.

Конечно, это не полный список доступных ресурсов, но именно эти сайты мы рекомендуем посетить в первую очередь. Информацию о совместимости с Linux устройства, которое вы хотите приобрести, можете также получить непосредственно на сайте производителя. Если производители позиционируют свою продукцию как совместимую с Linux, то у них на сайте должны быть доступные для скачивания драйверы или инструкции. Не забывайте также об огромном количестве справочной информации по Linux, которую вы можете найти с помощью поисковой системы.

Наконец, попытайтесь найти общество пользователей Linux (LUG — Linux User's Group) в стране своего проживания. LUG — это локальное сообщество людей, увлекающихся Linux и его усовершенствованием. Вы найдете разных людей с различным уровнем знаний: от системных администраторов, разработчиков и руководителей компаний до обычных пользователей Linux. Члены LUG регулярно встречаются, чтобы совместно обсудить проблемы и продемонстрировать свои пути усовершенствования Linux и связанные с этим технологии.

Некоторые LUG финансируют события местного масштаба, как, например, инсталл-фесты (http://en.wikipedia.org/wiki/Install_fest) или другие мероприятия по продвижению Linux. Скорее всего, если вы зададите вопрос на встрече членов LUG, кто-нибудь из них (и наверняка не единственный) будет знать ответ. Если вы решите присоединиться к местному LUG, то найти его сможете с помощью поиска в Интернете. У большинства LUG есть сайты или интернет-рассылки, которые без труда могут быть найдены в Сети.

Программное обеспечение для Ubuntu

Большинство программ для Ubuntu можно найти на сайте, содержащем пакеты программного обеспечения (http://packages.ubuntu.com). Наиболее распространенным способом установки программ на Ubuntu является использование стандартных приложений — Synaptic, APT и Update Manager (в гл. 2 подробно описан процесс поиска и установки программного обеспечения).

Возможно, когда-нибудь вы захотите поэкспериментировать и найти программы, не входящие в состав стандартных пакетов. Большинство таких пакетов будет содержать ключи MD5sum или GPG, чтобы вы могли определить, что они являются подлинными (www.debian-administration.org/articles/375). При этом вы также можете столкнуться с вопросами совместимости с нестандартным программным обеспечением, что может усложнить процесс выполнения обновлений. При работе с нестандартным программным обеспечением ключ необходим, чтобы можно было проверить программы и убедиться, что они не наносят вред системе. Ниже приведен список сайтов, на которых можно найти разнообразные программы.

ВНИМАНИЕ

Следует внимательно относиться к вопросу совместного использования оригинального программного обеспечения Ubuntu и программ из других источников и проверять подлинность загружаемого программного обеспечения.

- http://www.happypenguin.org — содержит архив игр для Linux. Здесь размещена нескончаемая коллекция игр для Linux: от экспериментальных демо-версий до коммерческих игр от компаний-разработчиков, предоставляющих исходные коды или бинарные файлы, чтобы их игры могли быть совместимы с Linux (спасибо вам, коммерческие компании-разработчики игр!).
- http://www.freshmeat.net — этот сайт по праву гордится самой большой коллекцией программ, тем, заставок, а также программного обеспечения для Palm-OS, работающего в UNIX и UNIX-подобных системах. Для каждой программы здесь создан раздел, предназначенный для обсуждения, что облегчает сам процесс обсуждения и дает возможность обратной связи. Эти ребята активно работают уже очень долгое время.
- http://sourceforge.net — когда разработчики открытых кодов собираются вместе для запуска нового проекта, многие из них обращаются к ресурсу SourceForge, чтобы разместить здесь свой проект. SourceForge предоставляет как интернет-пространство, так и приложения, необходимые для управления проектами, ресурсами, коммуникациями и кодом. Если вы ищете программное обеспечение, обязательно загляните на этот сайт.
- http://www.linux.org/apps — данный сайт активно работает с 1994 года. Его появление было связано с потребностью размещения информации о Linux. Этот сайт является постоянно пополняющимся источником информации относительно всего, что касается Linux. Раздел, посвященный приложениям, содержит ссылки на большое количество разнообразного программного обеспечения для операционных систем Linux.

Описание команд для Linux

Сегодня многие важные задачи в Linux можно выполнять как через графический интерфейс, так и с помощью команд, однако командная строка всегда была и остается тем интерфейсом, который выбирают опытные пользователи Linux.

Графические интерфейсы пользователя (GUI) подразумевают под собой интуитивность. Имея некоторый опыт пользования компьютером, вы, вероятно, поймете, как, к примеру, добавить пользователя, изменить время и дату и настроить звуковую карту через GUI. В таких случаях мы просто будем называть, каким графическим приложением необходимо воспользоваться в ходе вашей работы. Однако в следующих случаях вам, вероятно, придется отдать предпочтение командной строке.

- **Почти всегда что-нибудь работает неправильно** — задать вопрос на онлайн-форуме или попросить помощи — все это осуществляется через командную строку. К тому же запросы, выполняемые через командную строку, являются значительно более продуктивными, если речь идет о проблеме с настройкой какого-нибудь устройства или получения доступа к файлам и справочникам.
- **Удаленное системное администрирование** — если вы администрируете удаленный сервер, вполне возможно, что графические приложения вам недоступны. Но даже если приложения удаленного доступа к GUI (X-приложения или VNC) и приложения, доступные через Сеть, будут вам доступны, они, как правило, работают значительно медленнее, чем командная строка.
- **Элементы, не поддерживаемые GUI,** — административные средства GUI способны предложить лишь самые общие пути выполнения задачи. Для выполнения более сложных операций часто требуются использовать параметры, доступные только при использовании командной строки.
- **GUI вышел из строя или не установлен** — если недоступна ни одна графическая оболочка или если установленный GUI работает неверно, вам придется работать с командной строкой. Некорректная работа GUI может быть вызвана рядом причин: например, если вы используете бинарный драйвер NVIDIA или ATI от стороннего разработчика, а обновление ядра делает драйвер несовместимым с системой.

Можно лишь добавить, что для того, чтобы раскрыть все возможности Linux, необходимо уметь пользоваться командной оболочкой. Для контроля и управления операционной системой Linux существуют тысячи команд, но, независимо от того, кем вы являетесь — опытным пользователем Linux или новичком, — одна проблема выглядит угрожающе: как вспомнить самые важные команды и параметры, когда командная оболочка выдает лишь следующее:

```
$
```

Данная книга представляет собой не просто очередной сборник команд или изложение MAN-справочника. Здесь команды описываются таким образом, чтобы вы могли их попутно использовать. Другими словами, вместо того, чтобы перечислять команды в алфавитном порядке, мы поместили описания команд, предназначенных для работы с файловыми системами, подключения к сетям и управления процессами, в отдельные главы книги. Таким образом, если вам понадобится найти какую-либо команду, то необязательно будет искать ее лишь по названию.

Точно так же мы не будем перечислять и все параметры, доступные для каждой команды, — вместо этого мы наглядно покажем, как работают самые важные

параметры той или иной команды, а затем продемонстрируем быстрые способы поиска дополнительных параметров на MAN-страницах, в информационной базе info или справке.

Поиск команд

Некоторые команды, описываемые в этой книге, могут не быть установлены по умолчанию в вашем дистрибутиве Ubuntu, но они наверняка будут доступны через APT или другие источники. Если вы попытаетесь выполнить команду, которую командная оболочка не сможет найти, то увидите сообщение об ошибке примерно следующего содержания:

```
$ sillycommand
-bash: sillycommand: command not found
```

Может быть несколько причин вывода на экран такого сообщения:

- была допущена опечатка при наборе текста команды;
- команда не находится ни в одном из каталогов, путь к которым указан командной оболочке;
- команда может быть доступна только для суперпользователя (с помощью использования команды sudo или другого подобного метода);
- команда или программный пакет Ubuntu, содержащий эту команду, просто не установлены.

В табл. 1.1 приведены команды, доступные для использования во всех дистрибутивах Linux, чтобы проверить, какие из введенных вами команд присутствуют в системе. В табл. 1.2 приведены те же команды, но применимые только к системам Ubuntu и Debian.

Если вы обнаружите команду на выходе поиска, написанную через apt-cache, или вам покажется, что нужная команда не установлена, то можете установить ее через Интернет, выполнив следующую команду:

```
$ sudo apt-get install название_пакета
```

Здесь название_пакета — название того программного пакета, который необходимо установить.

Таблица 1.1. Типичные команды Linux для поиска уже установленных команд

Команда и пример вывода	Описание
$ echo $PATH /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games	Показывает текущий путь
$ which mount /bin/mount	Отображает путь первого использования команды mount
$ find /usr -name umount /usr/lib/klibc/bin/umount	Ищет в файловой системе /usr файл или каталог с названием umount

Команда и пример вывода	Описание
$ whereis mount mount: /bin/mount /usr/share/man/man8/mount.8.gz	Отображает адрес первого бинарного файла и MAN-страницы для команды mount
$ locate mount ... /usr/bin/fdmountd	Использует команду locate для поиска списка папок (настраиваемых) для команды mount
$ apropos umount ... umount (8) - unmount file systems	Ищет в MAN-справочнике описание копий ключевого слова; в данном случае это umount
$ man 8 umount Reformatting umount(8), please wait...	Ищет umount в разделе 8 MAN-справочника (для выхода нажмите клавишу Q)

Таблица 1.2. Команды Ubuntu/Debian для поиска уже установленных команд

Команда и пример вывода	Описание
$ apt-cache search umount gnome-mount - wrapper for (un)mounting and ejecting storage devices --	Ищет кэшированные пакеты, которые могут содержать команду или описание для umount
$ dpkg-query -s umount ... initscripts: /etc/init.d/umountnfs.sh --	Ищет в установленных пакетах файл с именем umount, проверяя содержимое пакетов
$ dpkg -L initscripts ... /bin/mountpoint --	Отображает список всех файлов, содержащихся в пакете initscripts
$ sudo apt-get update Password: Get:1 http://security.ubuntu.com feisty-security Release.gpg [191B] --	Обновляет список кэшированных пакетов

Справочная информация по Ubuntu

Оригинальная документация по Linux и UNIX представлена в виде справочных страниц, которые обычно называются *MAN-страницами*. Документация, содержащая несколько более сложную информацию, появилась позже в виде информационной базы GNU *info*. Кроме того, для каждой конкретной команды почти всегда доступна справка (help messages — справочные сообщения).

Эта справочная информация ориентирована на различные компоненты. Для каждой команды, установленной на системе, существуют отдельные MAN-страницы. В справочнике также присутствует документация по устройствам, форматам

файлов, системе, а также информация о многих других компонентах Linux и разработчике. Документация, более подробно описывающая целые пакеты, находится в подпапках папки /usr/share/doc.

Ubuntu запаковывает большинство документации, поэтому, перед тем как ее прочесть, ее приходится распаковать. Для этого может использоваться программа gzip, которая не распаковывает файлы на жесткий диск, а лишь отображает их содержимое на экране монитора. Команда для распаковки документации, описывающей команду mount, выглядит следующим образом:

```
$ gzip -dc /usr/share/doc/mount/README.mount.gz
mount/umount for Linux 0.97.3 and later
...
```

MAN-справочник, информационная база info и папки /doc доступны на большинстве систем Linux.

Работа со справкой

Почти для всех команд в Linux доступно в случае необходимости что-то вроде краткого руководства по их применению. Зачастую, чтобы получить эту информацию, достаточно лишь ввести в команду аргумент -h или --help. Следующая команда позволяет получить руководство по применению команды ls:

```
$ ls --help
Usage: ls [OPTION]... [FILE]...
List information about the FILEs (the current directory by default).
...
```

Если информация, выводимая с помощью параметра --help, не помещается на экране, вы можете воспользоваться командой постраничного вывода для ограничения количества выводимой информации:

```
$ ls --help | more
...
```

ПРИМЕЧАНИЕ

Команда more является наиболее распространенной командой постраничного вывода, которую можно встретить на любой используемой сейчас системе UNIX. В системах Linux чаще используется команда less, которая, несмотря на странное название (досл. «меньше»), по иронии является более функциональной, чем more. Она позволяет пролистывать выводимые данные как вверх, так и вниз, использовать клавиши управления курсором для прокрутки, перемещаться по тексту и осуществлять поиск в нем с помощью тех же клавиш, что и в редакторе vi.

В предыдущих примерах было показано, как вывести справку команды ls на экран, однако, используя команду card, выводящую справку на печать через установленный по умолчанию принтер, вы можете также форматировать полученные справочные данные. Кроме того, эти данные могут быть сохранены в виде Postscript-файла, который впоследствии можно открыть любой программой для просмотра файлов или с помощью программы ps2pdf конвертировать в PDF-файл.

MAN-страницы

В табл. 1.1 был кратко изложен процесс использования команды apropos для поиска соответствующего раздела MAN-справочника, касающегося команды umount. Кроме того, команда apropos может использоваться для поиска нужной страницы справочника по ключевому слову или нескольким буквам. На выходе команды будут показаны MAN-страницы, содержащие введенное слово.

```
$ apropos crontab
/etc/anacrontab (5) [anacrontab] - configuration file for anacron
anacrontab (5)        - configuration file for anacron
crontab (1)           - maintain crontab files for individual users (V3)
crontab (5)           - tables for driving cron
```

В данном случае на выходе команды apropos показаны разделы и названия MAN-справочников, в которых было найдено слово crontab, при этом разделы MAN-справочников сгруппированы по темам. MAN-страницы раздела 1 относятся к теме «Выполняемые программы или команды для командной оболочки», раздела 5 — к теме «Форматы файлов и условные обозначения». Разделы MAN-справочника одинаковы для всех систем Linux, но могут немного отличаться на других UNIX-подобных системах. Выполнить запрос на поиск man в справочнике, чтобы определить, какие их разделы присутствуют на вашей системе, можно с помощью следующей команды:

```
$ man man
Reformatting man(1), please wait...
...
```

В табл. 1.3 приведены номера разделов справочника и их названия.

Таблица 1.3. Разделы MAN-страниц

Номер	Название раздела
1	Выполняемые программы или команды для командной оболочки
2	Обращение к операционной системе (функции, предоставляемые ядром)
3	Обращение к библиотеке (внутренние функции программных библиотек)
4	Специальные файлы (обычно можно найти в /dev)
5	Форматы файлов и условные обозначения типа /etc/passwd
6	Игры
7	Прочее (включая макропакеты и условные обозначения), например man(7), groff(7)
8	Команды по администрированию системы (обычно только для root)
9	Программы ядра [нетипичные]

Итак, мы видим, что введенное слово crontab встречается в разделе 1 («Выполняемые программы или команды для командного процессора»), а также в разделе 5 («Форматы файлов и условные обозначения»). Вводя номера разделов

в качестве аргумента команды man, можно просматривать данные разделы MAN-справочника.

```
$ man 5 crontab
Reformatting crontab(5), please wait...
CRONTAB(5)                                          CRONTAB(5)
NAME
         crontab - tables for driving cron
DESCRIPTION
         A crontab file contains instructions to the cron(8) daemon of the
         general form: "run this command at this time on this date".
...
```

Если опустить номер, будет отображена MAN-страница первого найденного командой man раздела. В следующем примере команда man переходит в раздел 1 справочника, в котором упоминается crontab.

```
$ man crontab
Reformatting crontab(1), please wait...
CRONTAB(1)                                          CRONTAB(1)
NAME
         crontab - maintain crontab files for individual users (V3)
...
```

Помимо номеров разделов, для выполнения других задач в команду man можно добавлять еще некоторые аргументы, примеры использования которых приведены в табл. 1.4.

Таблица 1.4. Параметры команды man

Параметр	Описание
man -a crontab	Последовательно отображает все разделы справочника man, в которых упоминается crontab
man 5 crontab	Отображает части раздела 5 справочника man, в которых упоминается crontab
man crontab -P more	Использует пораметр more для постраничного просмотра частей справочника man, содержащих crontab
man -f crontab	Эквивалент команды whatis
man -k crontab	Эквивалент команды apropos

Команда whatis — это еще одна утилита поиска в man. Ее отличие от команды apropos состоит в том, что она отображает лишь описание страницы MAN-справочника, содержащей указанное ключевое слово. Выполнение команды apropos для поиска команды route выводит результат в виде трех страниц справочника, в которых было обнаружено слово route:

```
$ apropos route
NETLINK_ROUTE (7)      - Linux IPv4 routing socket
route (8)              - show / manipulate the IP routing table
traceroute6 (8)        - traces path to a network host
```

Если же для поиска информации о команде `route` запустить команду `whatis`, на экран будет выведена документация только из раздела 8:

```
$ whatis route
route (8) - show / manipulate the IP routing table
```

Документы info

Для некоторых команд, форматов файлов, устройств и других компонентов Linux разработчики привели более подробное описание, включив эту информацию в базу данных info. Это описание выполнено в виде набора ссылок на справочные онлайн-ресурсы. Войти в базу данных info можно, просто выполнив команду `info` или отдельно открыв саму утилиту (для выхода из утилиты info используйте клавишу **q**).

```
$ info ls
```

Приведенная выше команда выводит информацию о команде `ls`. Перемещаться по программе info можно, используя клавиши ↑, ↓, ← и →, а также **Page Up** и **Page Down**. В табл. 1.5 приведены сочетания клавиш, предназначенные для навигации в утилите info.

Среди пакетов программ, особенно подробное описание которых содержится в базе данных info, можно назвать gimp, festival, libc, automake, zsh, sed, tar, bash. Файлы, используемые info, находятся в каталоге `/usr/share/info`.

Таблица 1.5. Навигация в программе info

Клавиши	Действие
?	Отобразить основные команды, используемые в утилите info
Shift+L	Возвратить к предыдущему просмотренному элементу
N, P, U	Перейти к следующему, предыдущему элементу или вверх соответственно
Enter	Перейти по гиперссылке, находящейся под курсором
Shift+R	Перейти по перекрестным ссылкам
Q или Shift+Q	Завершить работу info и выйти из приложения

Резюме

В одной короткой главе мы описали некоторые сходства и различия между Ubuntu и другими дистрибутивами Linux и UNIX-подобными системами. Вы узнали про некоторые онлайн-ресурсы, касающиеся Linux вообще и Ubuntu в частности.

Вы также узнали, где можно найти программное обеспечение для Linux и, в частности, для Ubuntu. Используя Debian Advanced Package Tool (APT) , вы установили некоторые пакеты программ и увидели, где в системе можно найти нужные команды или страницы MAN-справочника. Вы также смогли поработать со стандартными потоками ввода и вывода stdin и stdout, перенаправляя вывод

команды (stdout) во временные файлы, а входящие потоки (stdin) — в другие команды.

Конечно, вы можете, если хотите, прочитать эту книгу от корки до корки, но она была написана в первую очередь с целью создать справочник, содержащий сотни команд для Ubuntu и Debian Linux, которые наиболее полезны продвинутым пользователям и системным администраторам. Поэтому, поскольку информация в книге организована по темам, а не в алфавитном порядке, нет необходимости заранее знать команду, чтобы найти ее в книге и выполнить необходимую операцию с ней.

Большинство описанных в книге команд с одинаковым успехом работают во всех операционных системах на базе Linux, а многие также совместимы с системами UNIX прежних версий.

2 Установка Ubuntu и программного обеспечения

APT (Advanced Package Tool) и dpkg (Debian package) являются проверенными временем приложениями, используемыми по умолчанию для установки программного обеспечения и последующего управления им. Это стандартные утилиты по управлению пакетами программ в Ubuntu и других системах, основанных на Debian, которые могут работать и в графическом режиме через самые простые GUI. Эти программы работают с DEB-файлами из интернет-репозиториев или DEB-файлами, хранящимися на жестком диске.

Эта глава освещает наиболее важные вещи, которые вам необходимо знать, чтобы установить Ubuntu. В ней также есть информация, касающаяся интернет-репозиториев Ubuntu. Ниже приведены подробные примеры практического использования APT и dpkg в виде командных строк.

риобретение и установка Ubuntu

Ubuntu и ее сестры Kubuntu, Xubuntu и Edubuntu оформлены таким образом, чтобы пользователю было максимально просто в них освоиться и начать работать. Эти дистрибутивы направлены на то, чтобы предотвратить появление пробелов в знаниях в ходе изучения новой операционной системы.

Программа-установщик Ubuntu (Ubiquity), которая сводит процесс установки к десяти щелчкам кнопкой мыши, является своеобразным введением в простой мир операционной системы Ubuntu. Компания Canonical Ltd, спонсирующая Ubuntu, облегчила даже сам процесс получения установочного CD, который можно бесплатно заказать по почте (не уплачивая даже налоги) (https://shipit.ubuntu.com/login). Если же вы являетесь обладателем высокоскоростного доступа в Интернет, то можете скачать с любого из многочисленных сайтов один из множества образов ISO (www.ubuntu.com/getubuntu/downloadmirrors). Список сайтов очень велик, что

обусловлено желанием предоставить людям необходимое количество серверов, с которых они могли бы скачивать систему. Если один сайт недоступен или не найден, попробуйте другой.

Сайты, с которых доступно скачивание Ubuntu, могут быть несколько непонятны, поэтому существует также расширенная страница загрузки (www.ubuntu.com/getubuntu/download), которая призвана помочь прояснить некоторые вещи. На момент написания книги последней предлагаемой версией Ubuntu являлась Ubuntu 7.04 (Feisty Fawn), а к октябрю 2007 года ожидалась Ubuntu 7.10, поэтому, возможно, вам уже будет доступна более новая версия. Расширенная страница загрузки предоставляет на сегодняшний день установочные пакеты, в которых есть возможность выбора между установкой системы на персональный компьютер и сервер. Вариант установки на сервер предназначен для людей, которым не нужен весь предлагаемый комплект приложений для персонального компьютера.

- Ubuntu 7.04 — наиболее стабильная на сегодняшний день версия Ubuntu. Именно ее выбирают чаще всего.
- Ubuntu 6.06 LTS — сопровождается длительной технической поддержкой (The Long Term Support), предоставляемой тем, кто оплатил трехлетнюю (для персональных компьютеров) или пятилетнюю (для серверов) поддержку.

Ниже описаны другие варианты системы, которые доступны на расширенной странице загрузки.

- **Standard personal computer** (Стандартный персональный компьютер) — этот параметр является стандартным для большинства пользователей. 64-битные процессоры AMD и Intel для настольных компьютеров и ноутбуков будут работать в случае, если отсутствуют специальные требования для оперативной памяти или приложений.
- **64-bit AMD and Intel computers** (Компьютеры на базе 64-битных процессоров AMD и Intel) — этот параметр следует выбрать в случае, если при работе с приложениями на компьютере необходимы большие объемы оперативной памяти или в особенности 64-битная платформа.
- **Sun UltraSPARC-based** — этот вариант Ubuntu предназначен для компьютеров на платформе Sun Micro-systems UltraSPARC RISC и является прекрасной альтернативой Sun Solaris (отлично работает на Ultra 60).

При выборе параметра **Alternate Desktop CD** (Альтернативный компакт-диск для настольных компьютеров) вы получите очень полезный набор утилит, например Logical Volume Management (LVM) (эта программа подробно описана в гл. 7). Если вам нужна LVM, то установите соответствующий флажок и лишь затем нажмите кнопку **Download** (Загрузить).

После того как загрузка будет завершена, вы сможете просмотреть указанный выше список сайтов и загрузить файл `MD5SUM` для скачиваемой версии Ubuntu (https://help.ubuntu.com/community/HowToMD5SUM). Он может помочь вам проверить подлинность образа ISO. Для большинства программного обеспечения с открытым исходным кодом также доступна цифровая подпись, и мы рекомендуем проверить ее перед установкой или записью образа на CD или DVD.

ПРИМЕЧАНИЕ

Если вы хотите обеспечить большую надежность загружаемых файлов, чем при использовании ключей MD5, попробуйте воспользоваться SecureApt. Для получения более подробной информации об использовании APT цифровой идентификации и шифрования посетите раздел SecureApt справочного сайта Ubuntu (https://help.ubuntu.com/community/SecureApt).

Подготовка к установке

Если вы перед установкой Ubuntu собираетесь удалить всю информацию с жесткого диска, то вам не требуется производить никаких предварительных операций. Если вам все же нужно сохранить некоторые данные, создайте их резервную копию. Чтобы установить Ubuntu, не удаляя никаких данных с жесткого диска, вам может понадобиться изменить размер локальных дисков и перераспределить место на жестком диске, создав новые разделы. Для получения полной информации, касающейся изменения размера локальных дисков и команд по созданию разделов, обратитесь к гл. 7.

Выбор параметров установки

После запуска стандартного установочного компакт-диска откроется меню с параметрами, которые представлены в табл. 2.1.

Таблица 2.1. Параметры установки Ubuntu (стандартный компакт-диск)

Параметр установки	Описание
Start or Install Ubuntu (Запустить или установить Ubuntu)	Начинает процесс установки
Start Ubuntu in safe graphics mode (Запустить Ubuntu в безопасном графическом режиме)	Загружает систему с компакт-диска, не используя при этом жесткий диск (применяется, если нужно что-то отладить или вы хотите что-либо изменить в структуре жесткого диска)
Install with driver update CD (Установить, используя компакт-диск с драйверами)	Процесс установки Ubuntu прекратится, и будет предложено вставить специальный компакт-диск с драйверами, чтобы продолжить установку
Check CD for defects (Проверка компакт-диска на наличие ошибок)	Проверяет компакт-диск, чтобы выявить проблемы заранее, а не в процессе установки
Memory test (Тест памяти)	Если у вас есть подозрения на наличие неполадок оперативной памяти, Ubuntu может предложить провести Memtest86 (www.memtest.org). Эта программа проведет тестирование вашей оперативной памяти с возрастающей нагрузкой, чтобы выявить в ней ошибки
Boot from first hard disk (Загрузка с первого жесткого диска)	Если вы случайно начали загрузку с компакт-диском в приводе, просто щелкните кнопкой мыши на этом пункте меню, и загрузка будет выполнена с первого жесткого диска

На альтернативном установочном CD (Alternate CD) нет ни безопасного графического режима, ни режима Driver update CD (Обновить драйверы с компакт-диска). Альтернативные им параметры описаны в табл. 2.2.

Таблица 2.2. Параметры установки Ubuntu (альтернативный компакт-диск)

Параметр установки	Описание
Install in text mode (Установка в текстовом режиме)	Установка производится с помощью текстовой консоли (используется, когда подсистема VGA недоступна или ее использование нежелательно)
Text mode install for manufacturers (Установка в текстовом режиме для производителей)	Устанавливает Ubuntu через пользовательскую учетную запись oem (original equipment manufacturer — самостоятельный производитель оборудования) для индивидуальной настройки системы. После запуска команды удаления учетной записи oem система при следующей загрузке попросит пользователя создать новую учетную запись
Install a command-line system (Установить систему с базовым набором консольных приложений)	Устанавливает Ubuntu только с основными приложениями (серверы, сетевые экраны, межсетевые интерфейсы, приложения с низким потреблением системных ресурсов)

Более подробную информацию о способах установки Ubuntu вы можете найти на странице Ubuntu wiki (https://wiki.ubuntu.com/Testing/InstallMethods).

Ответы на вопросы программы установки

После запуска установочного диска с Ubuntu чаще всего задают вопрос: «Ну и что дальше?» Поскольку загрузка Ubuntu, как правило, производится из Интернета в виде файла-образа Live CD, значит, вы можете запустить Ubuntu с компакт-диска без ее установки на жесткий диск. В этом случае следующим после запуска Ubuntu шагом является попытка поработать в системе. Если же вы решите установить Ubuntu, нажмите кнопку **Install** (Установить) на **Рабочем столе**, после чего будет запущена программа установки.

Большинство окон программы-установки выполнены в полностью понятной форме. В табл. 2.3 представлен краткий обзор этих экранов, а также некоторые советы, которые могут вам помочь в определенных случаях.

Таблица 2.3. Краткий обзор окон программы установки Ubuntu

Название окна	Описание
Install welcome (Вас приветствует программа установки)	Выберите нужный язык
Where are you? (Где вы находитесь?)	Укажите свое месторасположение для определения временнóй зоны
Keyboard layout (Раскладка клавиатуры)	Выберите используемую вами раскладку клавиатуры

Название окна	Описание
Prepare disk space (Подготовьте дисковое пространство)	Если хотите автоматически разбить жесткий диск на разделы, выберите Guided partitioning (Автоматическое разбиение). Если же вы хотите самостоятельно определить способ разбиения диска на разделы, выберите вариант Manual (Вручную) (для работы с LVM вам понадобится альтернативный установочный диск — Alternate install CD)
Migrate Documents and Settings (Перемещение документов и установок)	Этот параметр может помочь вам сохранить важную информацию (и учетные записи пользователей) Windows и перенести их в Ubuntu
Who are you? (Кто вы?)	Введите имя пользователя, имя учетной записи, пароль и имя компьютера
Ready to install (Готовность к установке)	Нажмите кнопку Advanced (Дополнительные параметры), если хотите определить параметры загрузчика и участвовать в сборе информации. Для завершения установки нажмите кнопку Install (Установить)

Если вы в целях безопасности выполнили загрузку с **Рабочего стола** Ubiquity, то у вас должно быть шесть вариантов терминальных сессий, переключаться между которыми можно с помощью сочетаний клавиш **Ctrl+Alt+F#** (от **F1** до **F6**), необходимых для вызова командной консоли. Кроме того, для отображения информации об ошибках в ходе установки вы можете воспользоваться сочетанием клавиш **Ctrl+Alt+F8**. Для возврата к **Рабочему столу** Ubiquity нажмите сочетание **Ctrl+Alt+F7**.

Работа с программными пакетами Debian

Если для установки программного обеспечения вы предпочитаете использовать приложения GUI, то с **Рабочего стола** или из сессии ssh (при использовании параметра `-X ssh` туннеля X11 (см. гл. 13)) доступна программа Synaptic Package Manager. Утилита aptitude является графической оболочкой APT, которая способна работать как в текстовом, так и в командном режимах. Оболочка утилиты dpkg — dselect (**www.debian.org/doc/manuals/dselect-beginner**) — также доступна на большинстве систем, выполненных на основе Debian, но при ее изучении могут возникнуть трудности.

Для получения более подробной информации о форматах программных пакетов Debian ознакомьтесь с руководством Debian Programmers Manual (**www.debian.org/doc/manuals/programmer**) и Debian FAQ (**www.debian.org/doc/FAQ/ch-pkg_basics.en.html**).

В программных пакетах Ubuntu используется формат Debian (обычно архив AR), который является стандартным способом архивирования программного обеспечения для операционных систем на базе Debian. Поскольку компоненты программного обеспечения помещаются в отдельные программные пакеты Debian (с расширением DEB), программное обеспечение может не только представлять собой самостоятельные архивы с программами, но и предоставлять большое количество информации о содержимом пакета. Эти метаданные могут включать в себя

информацию о программном обеспечении, взаимозависимости, архитектуре компьютера, продавце, размере, лицензии и др.

После установки основных компонентов Ubuntu вы можете добавлять, удалять и производить другие операции по управлению файлами DEB для повышения удобства использования системы. Ubuntu, Kubuntu, Xubuntu, Edubuntu и большинство других операционных систем на базе Debian используют файлы DEB для установки остальных внутренних компонентов системы. Приложение aptitude очень удобно использовать для повседневных нужд по управлению программным обеспечением, однако для управления программными пакетами существует множество других приложений, и, возможно, вам понадобится время от времени их использовать.

- APT — используется для загрузки и установки программных пакетов из интернет-репозиториев. Команды APT (`apt-get`, `apt-cache` и т. д.) можно использовать также и для установки программных пакетов из папок, хранящихся на локальном диске, но все же чаще эта программа используется для работы с программным обеспечением, находящимся в Интернете.
- dpkg — применяется для работы с DEB-файлами, находящимися на CD или других запоминающих устройствах. Команда `dpkg` предлагает параметры настройки, установки и получения информации о системном программном обеспечении.
- aptitude — используется при работе в командной консоли с интернет-репозиториями. Эта программа рекомендуется в первую очередь, поскольку автоматически выполняет некоторые операции, которые в противном случае при работе с dpkg или APT вам придется совершать вручную.

В этой главе представлены разделы, в которых описывается каждая из этих утилит, причем мы выделили наиболее характерные случаи использования этих приложений.

ПРИМЕЧАНИЕ

Для получения более подробной информации об этих приложениях посетите страницу APT HOWTO (www.debian.org/doc/manuals/apt-howto/ch-basico.en.html#s- dpkg-scanpackages) или прочтите разделы MAN-руководства, касающиеся APT и dpkg.

Ubuntu (и другие дистрибутивы *buntu) устанавливается с одного компакт-диска или DVD. После завершения установки системы для получения отчета об общем количестве доступных программных пакетов вы можете запустить команду `apt-cache stats`:

```
$ apt-cache stats
Total package names : 27748 (1110k)
 Normal packages: 21182
...
```

Как можно заметить, для базового варианта Ubuntu существует более 20 000 наименований программ, доступных в Сети. Сообщество Debian очень тщательно отбирает программное обеспечение, чтобы включать в пакеты лишь то, которое подходит для перераспределения. В инструкции Debian на сайте **www.debian.org/doc/manuals/debian-tutorial/ch-introduction.html** отмечено следующее: «Хотя Debian

придает большое значение распространению бесплатного программного обеспечения, бывает, что кому-то хочется (или необходимо) установить на компьютер частное (запатентованное) программное обеспечение. Сообщество Debian учитывает и по мере своих сил поддерживает это. Поэтому, несмотря на то, что платные программы не включаются в основные дистрибутивы системы, их иногда можно найти на ftp-сайтах в платных разделах. Кроме этого, существует постоянно растущее количество пакетов, созданных именно для того, чтобы устанавливать частное программное обеспечение, которое мы не имеем права самостоятельно распространять».

Компания Canonical придерживается похожих стандартов относительно Ubuntu (www.ubuntu.com/community/ubuntustory/licensing), предлагая программное обеспечение четырех категорий (www.ubuntu.com/community/ubuntustory/components).

- Main — программное обеспечение, которое свободно распространяется и поддерживается командой Ubuntu. Бо́льшая часть этого программного обеспечения устанавливается при установке Ubuntu.
- Restricted — программное обеспечение, общее для всех операционных систем на базе Linux, поддерживаемое командой Ubuntu, но необязательно имеющее абсолютно свободную лицензию.
- Universe — практически все свободное программное обеспечение, доступное в мире Linux, а также доступное в соответствии с лицензиями, которые не обязательно являются свободными, как остальные. Программное обеспечение этой категории не обеспечено полной безопасностью и технической поддержкой.
- Multiverse — программы, которые не соответствует концепции свободного программного обеспечения, являющейся основным компонентом лицензионной политики Ubuntu. Программному обеспечению этой категории никогда не оказывается техническая поддержка, а определение его лицензионной ценности — ваше право.

Более подробную информацию о компонентах программного обеспечения Ubuntu вы можете найти на сайте www.ubuntu.com/community/ubuntustory/components.

спользование программных пакетов

Ниже описываются основы работы с программными пакетами. Здесь рассказывается о процессах, скрытых от пользователя, и о том, как устанавливать пакеты. Изучение этих основ является важным первым шагом перед началом освоения работы других приложений, как, например, aptitude.

Команда dpkg является очень мощным инструментом, предназначенным для установки одиночных программных пакетов DEB, но она не обладает полным спектром возможностей и взаимозависимостей, которые необходимы для работы с различными программами, а также не может работать с репозиториями, как вышеперечисленные компоненты Ubuntu. С другой стороны, APT разрешает и устанавливает зависимости и работает с настроенными репозиториями, но не может быть использован, для установки файлов DEB, хранящихся на жестком диске.

Что делать, если появилось сообщение о локальной ошибке

Если вы работаете в командной строке Ubuntu (Feisty Fawn), то при попытке установить программные пакеты на экране может появиться сообщение о локальной ошибке, например одно из нижеперечисленных:

```
perl: warning: Setting locale failed.
perl: warning: Please check that your locale settings:
locale: Cannot set LC_CTYPE to default locale: No such file or directory
```

Проблема, скорее всего, связана с используемыми языковыми параметрами или международными кодировками вообще. Чтобы решить эту проблему, можно пойти по обходному пути: экспортировать переменную среды LC_ALL и приравнять ее к языковым настройкам (LANG setting):

```
$ export LC_ALL="$LANG"
```

На сайтах поддержки предлагаются еще некоторые способы решения этой проблемы, но данный путь является простейшим. Это решение должно работать в любом случае, вне зависимости от используемого вами языка, но помните, что вам нужно будет выполнять эту команду каждый раз при открытии локальной командной оболочки или оболочки ssh. Тем не менее, поместив команду в файл ~/.bashrc, можно автоматизировать эту задачу.

Некоторые другие дистрибутивы Linux для работы с программными пакетами также используют похожие на APT системы. Дистрибутивы, основанные на Red Hat или производные от них (включая CentOS, Fedora и Mandriva), для управления программным обеспечением используют такие приложения, как yum, rpm, urpmi и smart. Несмотря на то что все эти приложения значительно отличаются от тех, которые используются в Ubuntu, идея у них одна и та же: сначала устанавливается конфигурационный файл, указывающий приложению, где в Сети расположены самые свежие программные пакеты, а затем приложение в связке с программой установки загружает программное обеспечение в систему.

Такая система из программных онлайн-пакетов и внутреннего приложения по управлению ими является очень мощной комбинацией, легко снабжающей операционную систему последними достижениями, позволяющей разработчикам быстро и массово распространять внесенные ими изменения в программы, а также решать задачи взаимозаменяемости, цифровой идентификации и целостности программного обеспечения.

Открытие большого количества репозиториев для APT

В предыдущих версиях Ubuntu multiverse- и universe-репозитории по умолчанию были недоступны. Сейчас они по умолчанию открыты для Ubuntu, и, таким обра-

зом, обновление программ и их поиск могут осуществляться со значительно большим количеством параметров. Единственным недостатком является тот факт, что поддержка, лицензирование и файлы с исправлениями недоступны для universe- и multiverse-репозиториев. Это может быть проблемой, если при установке программного обеспечения вы придерживаетесь определенной политики и определенных процедур.

Чтобы отключить universe- или muliverse-репозитории, откройте в текстовом редакторе файл /etc/apt/sources.list и закомментируйте строки, имеющие включенные multiverse- или universe-компоненты. При желании, чтобы отметить закомментированные элементы, добавьте к строкам комментарии. В следующих примерах подпись отмечена знаком #cn:

```
#cn deb http://us.archive.ubuntu.com/ubuntu/ feisty universe
#cn deb-src http://us.archive.ubuntu.com/ubuntu/ feisty universe
#cn deb http://us.archive.ubuntu.com/ubuntu/ feisty multiverse
#cn deb-src http://us.archive.ubuntu.com/ubuntu/ feisty multiverse
#cn deb http://security.ubuntu.com/ubuntu feisty-security universe
#cn deb-src http://security.ubuntu.com/ubuntu feisty-security universe
#cn deb http://security.ubuntu.com/ubuntu feisty-security multiverse
#cn deb-src http://security.ubuntu.com/ubuntu feisty-security multiverse
```

Аналогично, если вы захотите добавить дополнительные репозитории, которые могут быть предложены частными лицами или компаниями, добавьте соответствующую строку в файл /etc/apt/sources.list. Для редактирования этого файла вы должны обладать правами суперпользователя:

```
$ sudo vi /etc/apt/sources.list
```

Добавьте строку, начав с deb (если речь идет об уже готовых пакетах) или deb-src (если речь идет об исходном коде), затем URL-адрес репозитория вместе с дистрибутивом (как feisty в предыдущем примере) и описания компонентов (примеры universe). Обычно компоненты описываются как contrib-предоставленные (то есть не имеющие отношения к проекту Ubuntu) и free (свободные) или non-free (несвободные). Обычно подобная информация содержится на сайте, предоставляющем репозиторий.

Если вы решите добавить непроверенные репозитории, убедитесь в подлинности и целостности предлагаемого программного обеспечения, прежде чем вносить изменения в свою операционную систему. Хотя сегодня это уже не является проблемой для пользователей Linux, по невнимательности все же очень просто добавить в систему некачественное или вредоносное программное обеспечение.

Используйте программное обеспечение только из проверенных источников и всегда проверяйте скачанные программы, перед тем как их установить. Для получения более подробной информации о репозиториях с программным обеспечением посетите страницу Debian Repository HOWTO (**www.debian.org/doc/manuals/repository-howto/repository-howto**).

Вот пример из одного документа HOWTO:

```
deb ftp://sunsite.cnlab-switch.ch/mirror/debian/ unstable main contrib non-free
```

Управление программным обеспечением с помощью APT

Хотя dpkg и APT могут работать в связке, одной программы APT вам будет вполне достаточно для работы в любой системе на основе Debian: загружать любые необходимые программы, обновлять их, проверять или выполнять поиск. В табл. 2.4 продемонстрированы варианты использования команд APT для выполнения различных задач. Для быстрого вывода справки относительно возможностей APT служит параметр -h командной строки.

ПРИМЕЧАНИЕ

Хотя для выполнения данных задач предпочтительнее использовать утилиту aptitude, а не APT, мы все же решили начать с описания последней, чтобы вы могли получить более основательные знания.

Таблица 2.4. Некоторые примеры использования Advanced Package Tool

Команда APT	Выполняемое действие
sudo apt-get update	Проверяет файл /etc/apt/sources.list и обновляет базу данных доступных программных пакетов. Выполняйте эту команду при каждом изменении sources.list
apt-cache search <ключевое слово>	Выполняет поиск по ключевому слову в базе данных программных пакетов (нечувствителен к регистру). Названия пакетов и их описания выводятся после нахождения ключевого слова
sudo apt-get install <package>	Загружает и устанавливает программный пакет по заданному названию, если таковой найден в базе данных. Начиная с APT 0.6, эта команда автоматически проверяет подлинность пакета, используя известные ей ключи gpg (http://wiki.debian.org/SecureApt)
sudo apt-get -d install <название программного пакета>	Загружает программный пакет и помещает его в папку /var/cache/apt/archives
apt-cache show <название программного пакета>	Отображает информацию о программах требуемого пакета
sudo apt-get upgrade	Проверяет наличие обновлений для всех установленных пакетов и предлагает загрузить и установить их
sudo apt-get dist-upgrade	Обновляет всю систему, даже если для этого требуется удаление пакетов программ. Примечание: этот метод обновления системы является нежелательным
sudo apt-get autoclean	Удаляет не полностью загруженные или еще не установленные пакеты
sudo apt-get clean	Удаляет из папки /var/cache/apt/archives все кэшированные пакеты для освобождения места на диске
sudo apt-get --purge remove <название программного пакета>	Удаляет указанный пакет и все конфигурационные файлы. Для сохранения конфигурационных файлов удалите из команды ключевое слово --purge

Команда APT	Выполняемое действие
sudo apt-get -f install	Ищет неработающие программные пакеты и пытается отладить те, которые отмечены сообщением «unmet dependency» (взаимозависимости не удовлетворены)
apt-config -V	Выводит информацию о версиях установленных приложений APT
sudo apt-key list	Отображает ключи gpg, известные APT
apt-cache stats	Выводит статистику по всем установленным пакетам
apt-cache depends	Выводит взаимозависимости программного пакета (установленного или нет)
apt-cache pkgnames	Выводит список всех установленных пакетов

В качестве примера мы установим бесплатную программу Google Picasa (**http://picasa.google.com**), разработанную компанией Google и предназначенную для управления цифровыми фотографиями и предоставления их для общего просмотра. В этом примере будет установлено программное обеспечение из непроверенного репозитория, использованы некоторые приложения APT для проверки подлинности и качества загружаемой программы и установлена сама программа.

Добавление непроверенных репозитория и ключа электронно-цифровой подписи

Чтобы приступить к использованию репозитория Google, откройте файл /etc/apt/sources.list в текстовом редакторе (nano, vi), используя команду sudo:

```
$ sudo vi /etc/apt/sources.list
```

Затем в начало файла sources.list добавьте следующие две строки:

```
# cn - added for google software
deb http://dl.google.com/linux/deb/ stable non-free
```

Кроме того, нужно **загрузить ключ электронно-цифровой подписи Google для подтверждения подлинности программных пакетов Google**. Этот цифровой ключ может быть загружен с помощью команды wget и помещен в папку /tmp, которая далее будет использована для хранения загружаемых файлов.

```
$ wget https://dl-ssl.google.com/linux/linux_signing_key.pub -O /tmp/key.pub
--08:26:46-- https://dl-ssl.google.com/linux/linux_signing_key.pub
'/tmp/key.pub'
...
```

Команда wget (описанная в гл. 12) загружает файл с сайта Google и помещает его в папку /tmp/key.pub. Здесь важно, чтобы данный файл являлся общим зашифрованным ключом, используемым для проверки пакетов, загружаемых с сайта Google.

Затем, используя команду apt-key, импортируйте ключ в APT:

```
$ sudo apt-key add /tmp/key.pub
Password:
OK
```

Чтобы убедиться, что цифровая подпись Google была загружена полностью (без недостающих данных), проверьте через APT ключи безопасности:

```
$ sudo apt-key list
...
uid              Google, Inc. Linux Package Signing Key
<linux-packages-keymaster@google.com>
sub 2048g/C07CB649 2007-03-08
```

Затем, чтобы обновить новый репозиторий, **обновите кэш пакетов APT**. Для этого используйте команды sudo и apt-get update. Проверьте репозиторий Google следующим образом:

```
$ sudo apt-get update
Get:1 http://dl.google.com stable Release.gpg [189B]
Ign http://dl.google.com stable/non-free Translation-en_US
Get:2 http://dl.google.com stable Release [1026B]
...
```

Поиск программных пакетов

Теперь, когда новый репозиторий установлен, вы можете выполнить **запрос на наличие нового программного обеспечения**:

```
$ apt-cache search picasa
picasa - Picasa is software that helps you instantly find, edit and share all the
pictures on your PC.
```

Кроме того, вы можете выполнить информационный запрос о данном пакете Picasa **через APT**:

```
$ apt-cache show picasa
Package: picasa
Version: 2.2.2820-5
...
```

Сколько дополнительного программного обеспечения потребуется для обновления Picasa? **Проверить взаимозависимости** можно следующим образом:

```
$ apt-cache depends picasa
picasa
 Depends: libc6
```

Установка пакетов

С помощью команды sudo, используя APT или другие приложения для работы с пакетами программ, вы уже можете **устанавливать любое доступное для Ubuntu программное обеспечение Google**. В следующем примере продемонстрирован процесс установки Picasa с помощью APT:

```
$ sudo apt-get install picasa
Reading package lists... Done
Building dependency tree
```

```
Reading state information... Done
The following NEW packages will be installed:
 picasa
0 upgraded, 1 newly installed, 0 to remove and 115 not upgraded.
Need to get 21.7MB of archives.
After unpacking 82.3MB of additional disk space will be used.
Get:1 http://dl.google.com stable/non-free picasa 2.2.2820-5 [21.7MB]
Fetched 21.7MB in 1m3s (340kB/s)
Selecting previously deselected package picasa.
(Reading database ... 88015 files and directories currently installed.)
Unpacking picasa (from .../picasa_2.2.2820-5_i386.deb) ...
Setting up picasa (2.2.2820-5) ...
```

Теперь вы можете запустить программу Picasa, щелкнув кнопкой мыши на ее названии в меню **Applications ▸ Graphics** (Приложения ▸ Графика) на **Рабочем столе** Ubuntu или выполнив в командной строке команду picasa.

Обновление пакетов

С течением времени программные пакеты изменяются, появляются новые их версии с улучшенными инструментами и исправленными ошибками. Для обновления своей системы до последней вы можете воспользоваться инструментами APT. Этот процесс осуществляется в два этапа.

Сначала проверьте наличие обновлений для программных пакетов вашей системы Ubuntu, применив параметр update apt-get:

```
$ sudo apt-get update
```

Эта команда производит в репозиториях поиск новых версий программных пакетов, доступных для скачивания, и обновляет список кэшированных пакетов и версий вашей операционной системы.

Затем обновите программные пакеты, используя параметр upgrade apt-get.

```
$ sudo apt-get upgrade
```

Перед тем как обновлять сами пакеты, вы всегда можете обновить список доступных пакетов, но возможно также успешно использовать обе эти команды одновременно. Если разделить эти команды точкой с запятой, они будут выполнены обе, одна за другой:

```
$ sudo apt-get update; sudo apt-get upgrade
Get:1 http://dl.google.com stable Release.gpg [189B]
...
The following packages will be upgraded:
 app-install-data app-install-data-commercial apport apport-gtk bind9-host capplets-
data dbus dbus-1-utils
...
112 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
Need to get 140MB of archives.
After unpacking 3891kB of additional disk space will be used.
Do you want to continue [Y/n]? n
```

Обновление одного программного пакета

Обновление отдельного системного пакета в Ubuntu является очень простой операцией, которую можно осуществить с помощью команды apt-get install <название программного пакета>. Данная команда должна выполняться через sudo. После ее выполнения старая версия автоматически изменяется на доступную новую.

ПРИМЕЧАНИЕ

Это может показаться достаточно странным, но параметр upgrade apt-get обновляет все пакеты. Параметр же install устанавливает новый пакет или обновляет один или более специальных пакетов.

Сначала **определите версию используемого программного обеспечения** (в данном примере определяется версия приложения minicom, отвечающего за передачу данных по линиям последовательной передачи). Как и в большинстве случаев, программа minicom поддерживает функцию вывода на монитор номера текущей версии:

```
$ minicom --version
minicom version 2.1 (compiled Nov 5 2005)
...
```

Теперь воспользуйтесь APT для установки из репозиториев последней версии пакета minicom. APT информирует о запуске процесса обновления:

```
$ sudo apt-get install minicom
...
The following packages will be upgraded:
 minicom
```

Теперь выполните запрос текущей версии программы minicom, и вы увидите, что вместо прежнего номера версии отображается новый:

```
$ minicom --version
minicom version 2.2 (compiled Mar 7 2007)
...
```

Удаление программных пакетов

С помощью параметра remove команды apt-get вы можете удалить пакет из операционной системы Ubuntu. Для этого вам нужно будет подтвердить удаление программного обеспечения:

```
$ sudo apt-get remove picasa
Reading package lists... Done Building dependency tree
Reading state information... Done
The following packages will be REMOVED:
 picasa
0 upgraded, 0 newly installed, 1 to remove and 115 not upgraded.
Need to get 0B of archives.
After unpacking 82.3MB disk space will be freed..
Do you want to continue [Y/n]? n
```

Очистка программных пакетов

Перед установкой новой версии Ubuntu все загруженные ранее программные пакеты кэшируются в папке /var/cache/apt/ для ускорения их закачки в случае, если они вам понадобятся в будущем. Со временем они могут начать занимать много места на диске. Вы можете удалить эти пакеты, но затем, если они потребуются для удовлетворения зависимостей, вам придется снова их загружать. Очистка кэша производится с помощью параметра clean команды apt-get. В следующем примере продемонстрирована работа этой команды. Сначала для определения имеющихся кэшированных пакетов применим команду find к папке /var/cache/apt/:

```
$ find /var/cache/apt/ -name \*.deb
/var/cache/apt/archives/picasa_2.2.2820-5_i386.deb
/var/cache/apt/archives/minicom_2.2-4build1_i386.deb
```

Теперь очистим кэшированные пакеты, находящиеся в папке APT, и проверим, все ли удалено, снова выполнив команду find:

```
$ sudo apt-get clean
$ find /var/cache/apt/ -name \*.deb
```

Теперь папка пуста.

ПРИМЕЧАНИЕ

Если вы используете параметр -h вместе с командой apt-get, то должны знать, что в версии APT, используемой в Ubuntu, применяется программа Super Cow Powers. Чтобы больше узнать об этом, выполните запрос moo в apt-get.

Управление программным обеспечением с помощью dpkg

Приложение dpkg работает на более низком уровне, чем утилиты APT, который лишь косвенно использует dpkg для управления программным обеспечением Ubuntu. В то же время APT и dpkg используют похожие принципы работы с утилитами YUM и RPM дистрибутивов на основе Red Hat Linux. Однако, хотя обычно возможностей APT хватает для выполнения практически всех необходимых операций, все же иногда необходимо использовать и dpkg (например, чтобы выяснить, какой программный пакет ассоциирован с конкретным файлом в вашей системе). В табл. 2.5 приведены некоторые основные команды и операции dpkg.

ПРИМЕЧАНИЕ

dpkg для обозначения отладочной информации, которую нужно вывести в ходе выполнения отдельных операций, использует параметр -D. Если же вам нужна более подробная информация, чем та, которая предлагается по умолчанию, попробуйте воспользоваться параметром -D1. В разделе man, посвященном dpkg, перечислены уровни выхода, используемые с меткой -D.

Таблица 2.5. Некоторые общие примеры использования утилиты dpkg

Команда dpkg	**Выполняемое действие**
dpkg -c <файл DEB>	Отображает файлы, установленные с помощью заданного DEB-файла (файл DEB должен быть указан в таком виде: путь к файлу/имя файла)
dpkg -I <файл DEB>	Выводит информацию о заданном DEB-файле
dpkg -p <название программного пакета>	Отображает информацию о заданном программном пакете
dkpg -S <имя файла>	Показывает пакеты, в которых был найден файл с заданным именем. Результатом выполнения команды может быть путь к файлу или просто его имя
dpkg -l	Отображает установленные пакеты, а также может предоставлять более подробную информацию о них
dpkg -L <название программного пакета>	Отображает все файлы, установленные из заданного программного пакета (пакет должен быть уже установлен)
dpkg -s <название программного пакета>	Выводит статус пакета
sudo dpkg -i <файл DEB>	Устанавливает указанный DEB-файл
sudo dpkg -r <название программного пакета>	Удаляет из системы указанный пакет, но оставляет файлы
sudo dpkg -P < название программного пакета>	Удаляет пакет вместе с конфигурационными файлами
sudo dpkg -x <файл DEB> <папка>	Распаковывает файлы, содержащиеся в DEB-файле, в указанную папку. После завершения операция восстановит права доступа к указанной папке

Команда dpkg позволяет любому пользователю выполнять запрос по базе данных программных пакетов, но, чтобы использовать эту команду для установки или удаления программного обеспечения со своего компьютера, необходимо обладать привилегиями суперпользователя.

Установка программного пакета

Команда dpkg работает только с пакетами, в то время как apt-get способна выполнять любые задачи, связанные с поиском в репозитории, в котором хранится необходимый программный пакет, а также загрузкой пакета из этого репозитория. Чтобы вам было проще разобраться, в следующем примере мы используем сначала команду apt-get для загрузки программного пакета, а затем dpkg для его установки, что позволит подчеркнуть отмеченную особенность команды dpkg.

Сначала с помощью apt-get загрузите файл DEB (параметр -d в команде необходим *только для загрузки*). Следующая команда **загружает, но не устанавливает пакет minicom**:

```
$ sudo apt-get -d install minicom
...
Fetched 168kB in 1s (131kB/s)
Download complete and in download only mode
```

Таким образом, вы получили DEB-файл для установки (данная команда загружает программный пакет в папку /var/cache/apt/archives).

Затем, используя команду dpkg -i (для *установки*), **установите загруженный DEB-файл**:

```
$ sudo dpkg -i /var/cache/apt/archives/minicom_2.2-4build1_i386.deb
Selecting previously deselected package minicom.
(Reading database ... 89127 files and directories currently installed.)
Unpacking minicom (from .../minicom_2.2-4build1_i386.deb) ...
Setting up minicom (2.2-4build1) ...
```

Удаление программного пакета

Чтобы **удалить установленный программный пакет** с помощью dpkg, воспользуйтесь параметром -r следующим образом:

```
$ sudo dpkg -r minicom
(Reading database ... 89182 files and directories currently installed.)
Removing minicom ...
```

Если вы решите удалить и пакет, и его конфигурационные файлы или удалить конфигурационные файлы после удаления самого пакета, выполните следующую команду:

```
$ sudo dpkg -P minicom
(Reading database ... 89126 files and directories currently installed.)
Removing minicom ...
Purging configuration files for minicom ...
```

Распаковка файлов из DEB-файла

Программные пакеты и Debian, и Ubuntu упаковываются в один DEB-архив. Каждый DEB-файл может содержать один и более файлов, из которых состоит соответствующий пакет. К этим файлам относятся предварительно подготовленная команда, файлы поддержки, документация и, возможно, исходный код. Таким образом, файл DEB, по существу, является архивом с файлами, которые устанавливаются на компьютер, а также некоторой важной и проверочной информацией, идентифицирующей программное обеспечение (описание, контрольные числа и т. д.). Используя команду dpkg, вы можете **извлечь всю эту информацию в программный пакет**. В следующем примере извлекаются файлы из DEB-архива, находящегося в заданной папке, в пакет rsync в папку /tmp:

```
$ mkdir /tmp/rsync_contents
$ sudo dpkg -x rsync_2.6.9-3ubuntu1.1_i386.deb /tmp/rsync_contents
$ ls /tmp/rsync_contents/
etc usr
```

Вы можете заменить файл rsync_2.6.9-3ubuntu1.1_i386.deb, используемый в этой команде, на любой загруженный вами архив DEB.

Сбор информации о программных пакетах DEB

Следующий пример показывает, как **среди установленных программных пакетов найти пакет rsync и вывести на экран монитора информацию о версии этого пакета**:

```
$ dpkg -p rsync
...
Version: 2.6.9-3ubuntu1.1
```

Для получения информации о DEB-архиве, находящемся в текущей папке, воспользуйтесь параметром -I:

```
$ dpkg -I rsync_2.6.9-3ubuntu1.1_i386.deb
new debian package, version 2.0.
...
```

Чтобы получить **список всех программных пакетов, установленных в системе**, выполните следующую команду:

```
$ dpkg -l | less
...
ii acpi 0.09-1 displays information on ACPI devices
...
```

Чтобы же просто **вывести информацию о конкретном пакете**, воспользуйтесь параметром -l и укажите название пакета:

```
$ dpkg -l rsync
...
ii rsync 2.6.9-3ubuntu1 fast remote file copy program (like rcp)
```

Проверьте, **относится ли заданный файл к программному пакету, и если относится, то к какому**:

```
$ dpkg -S /usr/bin/rsync
rsync: /usr/bin/rsync
```

Теперь, когда вы знаете, как выбрать программный пакет (или пакеты), запрос на который необходимо сформировать, можно перейти к изучению дополнительной информации. В следующем примере **показаны стандартные элементы установленного программного пакета**:

```
$ dpkg -s rsync
Package: rsync
Status: install ok installed
Priority: optional
Section: net
Installed-Size: 500
...
```

Ниже приведено **содержимое DEB-архива**, находящегося в локальной папке:

```
$ dpkg -c rsync_2.6.9-3ubuntu1.1_i386.deb
drwxr-xr-x root/root         0 2007-08-17 20:48 ./
```

```
drwxr-xr-x root/root         0 2007-08-17 20:48 ./usr/
drwxr-xr-x root/root         0 2007-08-17 20:48 ./usr/bin/
-rwxr-xr-x root/root    294864 2007-08-17 20:48 ./usr/bin/rsync
drwxr-xr-x root/root         0 2007-08-17 20:48 ./usr/share/
drwxr-xr-x root/root         0 2007-08-17 20:48 ./usr/share/doc/
...
```

В этом примере показан **процесс извлечения контрольных документов из DEB-архива на локальный диск в указанную папку**. Будьте внимательны при распаковке, так как эта команда устанавливает права доступа к указанной папке посредством значения 0755 (цифра 55 указывает, что все пользователи, за исключением тех, у кого уже были ограничены права доступа к папке /tmp, а также большинство приложений получат неограниченные права доступа к папке /tmp). В следующем примере будет создана папка вида /tmp/my_$RANDOM для работы в ней:

```
$ sudo dpkg -e rsync_2.6.9-3ubuntu1.1_i386.deb /tmp/my_$RANDOM
$ ls -lart /tmp/my_25445/
total 28
-rwxr-xr-x 1 root root 491  Aug 17 20:47 prerm
-rwxr-xr-x 1 root root 110  Aug 17 20:47 postrm
-rwxr-xr-x 1 root root 523  Aug 17 20:47 postinst
-rw-r--r-- 1 root root 37   Aug 17 20:48 conffiles
-rw-r--r-- 1 root root 926  Aug 17 20:48 control
drwxr-xr-x 2 root root 4096 Aug 17 20:48 .
```

Чтобы **извлечь все неуправляющие файлы**, содержащиеся в DEB-архиве, в папку, используйте параметр -x, как показано в следующем примере (здесь нужно знать, что права доступа к папке также будут возвращены в значение 0755):

```
$ sudo dpkg -x minicom_2.2-4build1_i386.deb /tmp/dx_$RANDOM
$ ls -lart /tmp/dx_4921/
total 16
drwxr-xr-x 4  root root 4096 Mar 7 09:10 usr
drwxr-xr-x 3  root root 4096 Mar 7 09:10 etc
drwxr-xr-x 4  root root 4096 Mar 7 09:10 .
drwxrwxrwt 11 root root 4096 Sep 1 08:19 ..
...
```

Чтобы **просмотреть установленные файлы программного пакета, используемого системой**, воспользуйтесь параметром -L:

```
$ dpkg -L minicom
/.
/usr
/usr/share
/usr/share/man
/usr/share/man/man1
/usr/share/man/man1/minicom.1.gz
/usr/share/man/man1/xminicom.1.gz
/usr/share/man/man1/ascii-xfr.1.gz
```

Если пакет не был полностью удален, вы увидите оставшиеся после него конфигурационные файлы:

```
$ dpkg -L minicom
/etc
/etc/minicom
/etc/minicom/minicom.users
```

Эти примеры описывают лишь стандартные случаи использования dpkg, поэтому приведенный перечень нельзя назвать исчерпывающим. Среди других доступных функций можно назвать следующие: изменение конфигурации (dpkg-reconfigure), выбор пакетов, которые следует игнорировать dpkg (dpkg hold), и определение параметров отбора. Для получения более полной информации ознакомьтесь с материалами, содержащимися на MAN-странице, посвященной dpkg.

Управление программным обеспечением с помощью aptitude

Приложения dpkg и APT применяются уже давно и хорошо себя зарекомендовали, но обе эти программы требуют от пользователя обладания достаточными знаниями, чтобы разбираться в приложениях и правильно с ними работать. Приложение aptitude упрощает работу, автоматизируя некоторые важные операции с программными пакетами (например, запуск команды apt-get update перед обновлением или установкой), и в то же время обладает достаточной гибкостью, чтобы быть полезным. Именно поэтому мы рекомендуем по возможности всегда использовать в командной строке именно данное приложение.

Главная цель программы aptitude — быть приложением одновременно и графической оболочки, и командной строки. В этом разделе будет подробно рассмотрено использование этого приложения в командной строке. В табл. 2.6 представлен анализ способов управления программным обеспечением с помощью aptitude. Стоит отметить, что большинство его параметров дублируют параметры команды apt-get.

ПРИМЕЧАНИЕ

Для получения более подробной информации об использовании графического интерфейса aptitude или других деталей, посетите страницу https://help.ubuntu.com/community/AptitudeSurvivalGuide (Aptitude Survival Guide) или http://people.debian.org/~dburrows/aptitude-doc/en/ (Aptitude user's manual) или выполните в командной строке команду man aptitude.

Таблица 2.6. Некоторые общие случаи использования aptitude

Команда	Выполняемое действие
sudo aptitude	Включает графический интерфейс. Для получения доступа к меню нажмите сочетание клавиш Ctrl+T, а для выхода — клавишу q
aptitude help	Отображает текстового помощника по работе в aptitude
aptitude search <ключевое слово>	Выводит пакеты, содержащие ключевое слово

Команда	Выполняемое действие
sudo aptitude update	Обновляет доступные индексы программных пакетов на основе информации, содержащейся в источниках APT
sudo aptitude upgrade	Обновляет все используемые пакеты до последних версий
aptitude show <название программного пакета>	Отображает информацию об указанном пакете (неважно, установлен он или нет)
sudo aptitude download <название программного пакета >	Загружает заданный пакет, но не устанавливает его
sudo aptitude clean	Удаляет все загруженные DEB-архивы из папки /var/cache/apt/archives
sudo aptitude autoclean	Удаляет все устаревшие DEB-архивы из папки /var/cache/apt/archives. Эта команда позволяет не допустить перезаполнения дискового пространства кэшированными файлами
sudo aptitude install <название программного пакета>	Устанавливает указанный программный пакет (для выбора конкретных версий и использования специальных символов могут использоваться различные параметры)
sudo aptitude remove <название программного пакета>	Удаляет из системы указанный пакет
sudo aptitude dist-upgrade	Обновляет все пакеты до самых последних версий, по необходимости удаляя или устанавливая пакеты. Параметр upgrade рекомендуется вводить через dist-upgrade

Обновление программных пакетов

По умолчанию aptitude всегда перед установкой или обновлением пакетов выполняет команду apt-get update, однако вы, если необходимо, все равно можете **выполнить только команду update**:

```
$ sudo aptitude update
Get:1 http://security.ubuntu.com feisty-security Release.gpg [191B]
Ign http://security.ubuntu.com feisty-security/main Translation-en_US
Get:2 http://us.archive.ubuntu.com feisty Release.gpg [191B]
...
```

Если вы хотите **обновить все пакеты системы**, то можете ввести параметр upgrade. В этом случае будут установлены все новые пакеты из репозиториев (в данном примере рассматривается ситуация, когда на жестком диске отсутствовали новые версии программных пакетов).

```
$ sudo aptitude upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Reading extended state information
Initializing package states... Done
Building tag database... Done
No packages will be installed, upgraded, or removed.
0 packages upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 0B of archives. After unpacking 0B will be used.
```

Сбор информации о программных пакетах

Используя **ключевые слова или полные названия пакетов, вы можете производить поиск** в aptitude точно так же, как и в других приложениях, предназначенных для работы с программными пакетами. Следующий пример демонстрирует поиск по ключевому слову «minic», с помощью которого можно отобразить информацию о пакете minicom и подключаемом расширении оконного менеджера xfce.

```
$ aptitude search minic
i minicom - friendly menu driven serial communication
program
p xfce4-minicmd-plugin - Mini-command line plugin for the Xfce4 panel
$ aptitude show minicom
Package: minicom
State: not installed
Version: 2.2-4build1
Priority: optional
Section: comm
```

Установка программных пакетов

Ранее вы уже загружали программный пакет, используя команду apt-get. Сейчас для этих же целей будет использована команда aptitude, которая позволяет **загрузить программный пакет, не устанавливая его**:

```
$ sudo aptitude download minicom
...
Get:1 http://us.archive.ubuntu.com feisty/main minicom 2.2-4build1 [168kB]
Fetched 1B in 0s (2B/s)
```

Если же вы хотите **установить пакет minicom**, то можете выполнить следующую команду:

```
$ sudo aptitude install minicom
...
Need to get 0B/265kB of archives. After unpacking 1401kB will be used.
Do you want to continue? [Y/n/?] n
```

Если у вас есть **несколько пакетов, которые нужно установить**, можете воспользоваться специальным символом. В данном случае будут установлены все программные пакеты, содержащие слово «minic» (как и в случае, описанном выше, когда производился поиск с помощью команды aptitude). Эта программа также определяет для каждого пакета все взаимозависимости, используя то, что на языке aptitude называется *matcher* («обнаружителем совпадений»). Чтобы установить все пакеты, содержащие слово «minic», используйте matcher ~n, поставив его в начале ключевого слова:

```
$ sudo aptitude install "~nminic"
...
The following NEW packages will be automatically installed:
 libxfce4mcs-client3 libxfce4mcs-manager3 libxfce4util4 libxfcegui4-4 lrzsz
```

```
xfce4-panel
The following NEW packages will be installed:
 libxfce4mcs-client3 libxfce4mcs-manager3 libxfce4util4 libxfcegui4-4 lrzsz
minicom xfce4-minicmd-plugin
 xfce4-panel
0 packages upgraded, 8 newly installed, 0 to remove and 0 not upgraded.
Need to get 702kB/967kB of archives. After unpacking 4645kB will be used.
Do you want to continue? [Y/n/?]
```

Удаление программных пакетов

Удалить программные пакеты с помощью aptitude даже проще, чем их установить. Просто попробуйте это сделать следующим образом:

```
$ sudo aptitude remove minicom

The following packages are unused and will be REMOVED:
lrzsz
The following packages will be REMOVED:
minicom
0 packages upgraded, 0 newly installed, 2 to remove and 0 not upgraded.
Need to get 0B of archives. After unpacking 1401kB will be freed.
Do you want to continue? [Y/n/?]
```

Очистка диска

Каждый раз, когда вы будете устанавливать программы с помощью команды aptitude, будет загружаться DEB-архив и помещаться в папку /var/cache/apt/archives. Через какое-то время, возможно, вам понадобится **удалить эти кэшированные файлы**, чтобы освободить место на диске. Для этого необходимо использовать параметр clean или autoclean. Если вы посмотрите, то увидите, что в этой папке уже есть кэшированные файлы:

```
$ ls /var/cache/apt/archives/
lock lrzsz_0.12.21-4ₐ1_i386.deb minicom_2.2-4build1_i386.deb partial
Removing these with aptitude only requires using the clean or autoclean option:
$ sudo aptitude clean
Reading package lists.  Done
Building dependency tree
Reading state information.  Done
Reading extended state information
Initializing package states.  Done
Building tag database.  Done
```

Повторное выполнение команды ls покажет, что пакеты на самом деле удалены, но если у вас низкая скорость интернет-соединения и вам необходима неделя, чтобы загрузить последние версии программных пакетов, то следует подумать дважды, прежде чем выполнять действия, описанные выше, либо воспользоваться параметром autoclean, который удаляет только устаревшие пакеты.

Полезные сочетания параметров aptitude

Параметр -v **расширяет возможности действий, выполняемых приложением aptitude**. Используя его много раз, вы можете получить больше, чем просто информацию, выводимую на экран после выполнения той или иной операции. Если вы выполните команду aptitude с параметром -v, то будет отображен ключ md5sum программного пакета — своего рода универсальный цифровой отпечаток пальцев, который можно использовать, если пакет был подделан или поврежден. **Использование параметра -vv позволяет получить еще более подробную информацию**:

```
$ aptitude show -vv minicom
Package: minicom
State: installed
...
Filename: pool/main/m/minicom/minicom_2.2-4build1_i386.deb
MD5sum: c408085cd37dfced2d3060b94ececd46
...
```

Чтобы еще **до выполнения операции имитировать ее результат**, можно воспользоваться параметром -s, который работает независимо от типа исполняемой команды.

```
$ sudo aptitude -s install minicom
Reading package lists... Done
...
Do you want to continue? [Y/n/?] y
Would download/install/remove packages.
```

Сочетание параметров -v и -s позволяет получить еще больше данных:

```
$ sudo aptitude -vs install minicom
Reading package lists... Done
...
Do you want to continue? [Y/n/?] y
Inst lrzsz (0.12.21-4.1 Ubuntu:7.04/feisty)
Inst minicom (2.2-4build1 Ubuntu:7.04/feisty)
Conf lrzsz (0.12.21-4.1 Ubuntu:7.04/feisty)
Conf minicom (2.2-4build1 Ubuntu:7.04/feisty)
```

Если вам **надоест каждый раз подтверждать выбор действия**, отвечая на вопрос «Хотите продолжить?», то можете ответить на него заранее, добавив параметр -y в исполняемую команду:

```
$ sudo aptitude -vs -y install "~ninc"
Reading package lists... Done
Building dependency tree
Reading state information... Done
Reading extended state information
Initializing package states... Done
...
```

Используя параметр -y, будьте предельно осторожны, поскольку команда aptitude не предусматривает возможности отката.

Наконец, параметр -h команды aptitude выводит отношение параметров, которое может быть использовано в любой момент, если требуется обновить информацию. Стоит отметить, что нам удалось найти недостаток в версии aptitude, используемой в Ubuntu:

```
$ aptitude -h
aptitude 0.4.4
...
This aptitude does not have Super Cow Powers.
```

Нет Super Cow Powers? Однако мы любопытны и спрашиваем aptitude с помощью параметра moo:

```
$ aptitude moo
There are no Easter Eggs in this program.
```

'Easter eggs (дословно «пасхальные яйца») представляют собой скрытые элементы программы. Хм. Может, попробуем воспользоваться параметром -v, чтобы получить более подробную информацию?

```
$ aptitude -v moo
There really are no Easter Eggs in this program.
```

Возможно, нужны еще более узкие параметры. Мы уже давим на aptitude, чтобы получить более подробную информацию:

```
$ aptitude -vv moo
Didn't I already tell you that there are no Easter Eggs in this program?
```

Тут уже начинает просматриваться некая система. Возможно, использование этих параметров в другом случае будет более успешным.

роверка установленных пакетов помощью программы debsums

Иногда возникают сомнения в качестве установленного на систему бинарного или открытого программного пакета. Они могут работать некорректно или вовсе не запускаться. Проблемы с поврежденными программными пакетами могут быть вызваны нестабильным интернет-соединением и перепадами в электропитании. Помимо этого, встречаются пользователи, которые могут попытаться заменить ключевые команды своими собственными, чтобы в дальнейшем наносить вред системе. Поэтому полезно сравнивать файлы файловой системы с информацией программного пакета.

Программа debsums является утилитой Ubuntu и других операционных систем на базе Debian, которая **сверяет информацию ключей MD5 каждого установленного пакета** с файлами md5sum, найденными в папке /var/ lib/dpkg/inf.

С помощью следующей команды вы можете установить эту программу:

```
$ sudo aptitude install debsums
```

В табл. 2.7 приведены наиболее полезные параметры команды debsums. Более подробную информацию о debsums можно получить в MAN-руководстве.

Таблица 2.7. Некоторые общие параметры утилиты debsums

Команда debsum	Выполняемое действие
debsums -a	Проверяет все файлы (включая конфигурационные, которые обычно располагаются слева)
debsums -e	Проверяет только конфигурационные файлы пакетов
debsums -c	Выводит список измененных на stdout файлов
debsums -l	Отображает файлы, в которых отсутствует информация о md5sum
debsums -s	Выводит только ошибки; при их отсутствии не производит никаких действий
debsums <название программного пакета>	Отображает пакеты, которые необходимо проверить с помощью debsums

ПРИМЕЧАНИЕ

Для выполнения большинства операций необязательно обладать правами суперпользователя (использовать команду sudo), однако к некоторым файлам простой пользователь не имеет доступа, поэтому, если появится сообщение «debsums: can't open at file /etc/at.deny (Permission denied)», вам придется использовать sudo.

Если выполнить команду debsums без дополнительных параметров, будет проверен каждый известный ей файл системы. При необходимости результат может быть переадресован в файл. Если md5sum проверяет выход для файла, название файла, выводимое утилитой debsums, сопровождается надписью OK, расположенной справа от сообщения. Кроме того, могут выводиться и другие сообщения: md5sums missing («md5sums потерян») для отдельного файла или слово REPLACED («Перемещен»), если утилита не нашла требуемого соответствия md5sum. Позже, если вы решите использовать это приложение в качестве основной программы получения нужной информации, чтобы все было установлено именно так, как вы хотите, вы сможете восстановить md5sum для отсутствующих или неработающих приложений. Таким образом, вы будете уверены, что у вас установлены наиболее свежие приложения.

Эта команда **сверяет все файлы системы с имеющимися в наличии файлами md5sum**. В приведенном ниже примере видно, что некоторые файлы отсутствуют или перемещены. Перед тем как восстановить файлы md5sum, убедитесь, что в системе отсутствуют проблемы с этими файлами:

```
$ debsums
/usr/bin/acpi                                              OK
/usr/share/man/man1/acpi.1.gz                              OK
/usr/share/doc/acpi/README                                 OK
/usr/share/doc/acpi/AUTHORS                                OK
...
/usr/share/app-install/icons/pybliographic.png             OK
debsums: no md5sums for bsdutils
debsums: no md5sums for bzip2
debsums: no md5sums for cdrecord
```

```
/usr/share/locale-langpack/en_AU/LC_MESSAGES/adduser.mo      REPLACED
/usr/share/locale-langpack/en_AU/LC_MESSAGES/alsa-utils.mo   OK
...
```

Если вы хотите сохранить эту информацию в файл, а также сохранить сообщения stdout и stderr, **переадресуйте stdout- и stderr-потоки в файл**. Чтобы впоследствии можно было продолжить работу в командной оболочке, в конце команды установлен знак &:

```
$ debsums &> /tmp/foo &
```

Чтобы **проверить конфигурационные файлы каждого программного пакета на наличие изменений**, выполните команду debsums с параметром -a:

```
$ debsums -a
/usr/bin/acpi                          OK
/usr/share/man/man1/acpi.1.gz          OK
...
```

Чтобы **проверить только конфигурационные файлы, пропуская остальные**, воспользуйтесь параметром -e. Данный способ позволяет убедиться, что вы не изменили случайно какой-либо конфигурационный файл. Ниже видно, что некоторые X-конфигурационные файлы были изменены:

```
$ debsums -e
...
/etc/X11/Xresources/x11-common                           OK
/etc/X11/Xsession                                        FAILED
/etc/X11/rgb.txt                                         OK
/etc/init.d/x11-common                                   OK
/etc/X11/Xsession.d/50x11-common_determine-startup       OK
/etc/X11/Xsession.d/30x11-common_xresources              OK
/etc/X11/Xsession.d/20x11-common_process-args            OK
/etc/X11/Xsession.options                                FAILED
...
```

По умолчанию debsums отображает достаточно большое количество информации, однако вы можете **отобразить и только измененные файлы**. Результат выполнения команды debsums с параметром -c показан ниже:

```
$ debsums -c
debsums: no md5sums for at
debsums: no md5sums for base-files
debsums: no md5sums for bc
...
```

Предыдущая команда отображает сообщения о файлах, содержащих информацию о md5sum. Выполнив команду debsums с параметром -l, вы сможете искать **файлы, которые не содержат информации о md5sum**:

```
$ debsums -l
at
base-files
bc
```

```
binutils
binutils-static
...
```

Если вы захотите, чтобы debsums **вывел информацию только об ошибках**, воспользуйтесь параметром -s:

```
$ debsums -s
debsums: no md5sums for at
debsums: no md5sums for base-files
debsums: no md5sums for bc
debsums: no md5sums for binutils
...
```

Чтобы **проверить конкретный пакет**, в качестве аргумента debsums укажите имя соответствующего пакета:

```
$ debsums coreutils
/bin/cat OK
/bin/chgrp OK
/bin/chmod OK
...
```

Следующая команда проверяет только те файлы, которые перечислены в файле md5sum пакета, содержащегося в папке /var/lib/dpkg/info, поэтому, если в программном пакете отсутствует файл md5sum, будет выведено сообщение об ошибке:

```
$ debsums rsync
debsums: no md5sums for rsync
```

Для генерации недостающих данных md5sum для rsync воспользуйтесь комбинацией из dpkg и утилиты md5sum и простым сценарием для командного процессора. Сначала для получения списка всех известных dpkg-файлов в пакете rsync выполните команду dpkg -L. В полученном списке будут, помимо имен файлов, содержаться другие строки данных. Далее необходимо будет передать полученные данные grep и отфильтровать все записи, которые не начинаются со знака /. Затем определите, чем является строка выводимых dpkg-данных — папкой или файлом (названия папок также начинаются с символа /). Если это файл, значит md5sum будет выполнена на строке вывода, которая на этом этапе будет являться именем файла. Наконец, сохраните все данные в текстовый файл с тем же соглашением об именах, что и файлы md5sum в папке /var/lib/dpkg/info.

```
$ for file in `dpkg -L rsync | grep ^/`; do
test -f "$file" && md5sum "$file";
done > /tmp/rsync.md5sums
```

Эта команда полезна тем, что позволяет получить базу данных md5sum, которая затем может быть записана на компакт-диск и использована для последующей проверки системы. Если файлы md5sum записаны на компакт-диск, они не могут быть случайно удалены или стать причиной возникновения проблем с файловыми системами жесткого диска. Если вы решите позже проверить ключи md5sum, то

воспользуйтесь командой md5sum вместе с параметром -c и добавьте к ней имя файла данных md5sum:

```
$ md5sum -c /tmp/rsync.md5sums

/usr/bin/rsync: OK
/usr/share/doc/rsync/examples/rsyncd.conf: OK
/usr/share/doc/rsync/README.gz: OK
/usr/share/doc/rsync/TODO.gz: OK
...
```

Чтобы воспользоваться файлом rsync.md5sum с приложением debsums, потребуется внести одно изменение, которое может повлечь за собой проблемы с md5sum, но все же является обязательным условием для работы с debsums, — убрать первый слэш в имени файла. Сделать это можно в текстовом редакторе либо командном процессоре:

```
$ cat /tmp/rsync.md5sums
302916114c29191cd9c8cb51d67ee60a /usr/bin/rsync
...
```

Чтобы убрать первый слэш в начале /usr/bin/rsync, попробуйте воспользоваться текстовым или потоковым редактором (Stream Editor):

```
$ sed -e 's# /# #g' /tmp/rsync.md5sums > /tmp/rsync.debsums
$ cat /tmp/rsync.debsums
302916114c29191cd9c8cb51d67ee60a usr/bin/rsync
```

Теперь, удалив первый слэш в rsync.debsums, вы можете скопировать этот файл в папку /var/lib/dpkg/info, после чего debsums сможет использовать его:

```
$ sudo mv /tmp/rsync.debsums /var/lib/dpkg/info/rsync.md5sums
$ debsums rsync
/usr/bin/rsync OK
/usr/share/doc/rsync/examples/rsyncd.conf OK
/usr/share/doc/rsync/README.gz OK
...
```

Создание DEB-архивов

Посредством реорганизации DEB-архива, используемого для формирования программного пакета Debian, вы можете изменять его, чтобы было удобнее пользоваться программным обеспечением (например, включив в архив файл md5sum). Для этого необходимо распаковать DEB-архив, который вы планируете изменить, в рабочую папку, после чего вы сможете изменять дерево файлов и управляющие файлы в соответствии со своими потребностями. Например, выполнив следующие команды, вы можете загрузить и распаковать пакет rsync и управляющие файлы в текущую папку (используемая в данном случае папка **$RANDOM** у вас будет, естественно, другой):

```
$ aptitude download rsync
```

Теперь извлеките из загруженного архива его содержимое и управляющие файлы (папку $RANDOM можно найти, введя /tmp/rsync_ и нажав клавишу Tab):

```
$ sudo dpkg -x rsync_2.6.9-3ubuntu1.1_i386.deb /tmp/rsync_$RANDOM
$ sudo dpkg -e rsync_2.6.9-3ubuntu1.1_i386.deb /tmp/rsync_17197/
```

Затем перейдите в папку с программными пакетами, в которую вы распаковали DEB-архив, и проверьте его содержимое. Вы должны будете найти папку, имеющую структуру, подобную следующей:

```
$ cd /tmp/rsync_17197
$ ls -lart
-rwxr-xr-x 1 root root 491 2007-08-17 20:47 prerm
-rwxr-xr-x 1 root root 110 2007-08-17 20:47 postrm
-rwxr-xr-x 1 root root 523 2007-08-17 20:47 postinst
drwxr-xr-x 4 root root 4096 2007-08-17 20:48 usr
drwxr-xr-x 4 root root 4096 2007-08-17 20:48 etc
-rw-r--r-- 1 root root 37 2007-08-17 20:48 conffiles
-rw-r--r-- 1 root root 985 2007-09-02 12:02 control
drwxr-xr-x 4 root root 4096 2007-09-02 12:02 .
drwxrwxrwt 10 root root 4096 2007-09-02 13:24 ..
```

Теперь необходимо настроить папку таким образом, чтобы согласовать форматы, требующиеся dpkg для формирования DEB-архива. Это предполагает создание подпапки rsync_2.6.9-3cn1.1/DEBIAN и перемещение в нее установочных файлов. Сам управляющий файл представляет собой особым образом форматированный файл, содержащий поле заголовка и поле содержимого (header field, content field). Он анализируется приложениями, предназначенными для работы с пакетами, для последующего вывода информации о пакете:

```
$ sudo mkdir -p rsync_2.6.9-3cn1.1/DEBIAN
$ sudo mv control conffiles prerm postrm postinst rsync_2.6.9-3cn1.1/DEBIAN
```

Вам также должны переместить папки etc/ и usr/ в папку rsync_2.6.9-3cn1.1:

```
$ sudo mv usr etc rsync_2.6.9-3cn1.1
```

Скорее всего, вы правильно завершили процесс добавления в архив и, если так, вся информация находится в подпапке rsync_2.6.9-3cn1.1 текущей папки.

Теперь переместите ранее созданный файл md5sum в подпапку DEBIAN и переименуйте ее (папку) в md5sums. Это укажет программе debsums путь к файлам md5sums, которые необходимо проверить:

```
$ sudo mv /var/lib/dpkg/info/rsync.md5sums rsync_2.6.9-3cn1.1/DEBIAN/md5sums
```

Теперь, чтобы изменить некоторую информацию, необходимо отредактировать управляющий файл. Естественно, вы не захотите устанавливать измененную версию rsync, содержащую оригинальную информацию о программном пакете. Откройте управляющий файл в редакторе vi или другом и измените строку Version так, как показано ниже. Если вы обратите внимание, то заметите, что после слова Version стоит двоеточие — это головное поле. Информационное поле следует сразу за ним. Убедитесь, что после двоеточия стоит пробел, и не добавляйте больше дополнительных пробелов (и не удаляйте их) в файле. Это является очень важным в процессе форматирования.

```
$ sudo vi rsync_2.6.9-3cn1.1/DEBIAN/control

...
Version: 2.6.9-3cn1.1
...
```

Немного ниже вы можете добавить поле Description. Оно будет отображаться в описании, когда потребуется узнать какие-либо детали программного пакета. Обратите внимание, что перед словами fast remote стоит пробел. Пробел является частью особого форматирования, на языке dpkg — текстом описания из многострочного заголовка. Если добавленное вами описание переносится на следующую строку, убедитесь, что в первом столбце стоит пробел:

```
...
Description: Modified by CN 2007-09-02 to include md5sums.
 fast remote file copy program (like rcp)
...
```

Теперь, используя команду dpkg -b и имя созданной вами для управляющего файла подпапки, создайте новый пакет. Программа предупредит вас, что Original-Maintainer является полем, определяющим пользователя. Можете не обращать на это сообщение внимания.

```
$ sudo dpkg -b rsync_2.6.9-3cn1.1
warning, `rsync_2.6.9-3cn1.1/DEBIAN/control' contains user-defined field
`Original-Maintainer'
dpkg-deb: building package `rsync' in `rsync_2.6.9-3cn1.1.deb'.
dpkg-deb: ignoring 1 warnings about the control file(s)
```

Теперь у вас есть новый DEB-архив, и вы можете через dpkg отобразить информацию о нем. Для этого просто выполните команду dpkg с параметром -I, и увидите информацию о пакете:

```
$ dpkg -I rsync_2.6.9-3cn1.1.deb
new debian package, version 2.0.
size 1004 bytes: control archive= 712 bytes.
    970 bytes,  21 lines     control
Package: rsync
Version: 2.6.9-3cn1.1
...
```

На данном этапе вы уже можете установить новый пакет rsync. Это упражнение демонстрирует в основном создание собственного программного пакета и не требует внесения изменений в систему, если отсутствует на то необходимость. Следующая команда показывает, что данный пакет будет установлен как обычный пакет Debian и будет обладать всеми его свойствами. Однако ведь вы хотите задействовать и debsums. Обратите внимание на то, что говорит dpkg:

```
$ sudo dpkg -i rsync_2.6.9-3cn1.1.deb
dpkg - warning: downgrading rsync from 2.6.9-3ubuntu1 to 2.6.9-3cn1.1.
(Reading database ... 88107 files and directories currently installed.)
Preparing to replace rsync 2.6.9-3ubuntu1 (using rsync_2.6.9-3cn1.1.deb) ...
Unpacking replacement rsync ...
Setting up rsync (2.6.9-3cn1.1) ...
```

Сейчас утилита debsums располагает некоторыми файлами для тестирования md5sum, поэтому вне зависимости от местоположения нового пакета rsync вы увидите следующее сообщение:

```
$ debsums rsync
/usr/bin/rsync                                  OK
/usr/share/doc/rsync/examples/rsyncd.conf       OK
/usr/share/doc/rsync/README.gz                  OK
...
```

Используя команду dpkg и параметр -l, вы можете просмотреть информацию о пакете и убедиться, что установлена его новая версия:

```
$ dpkg -l rsync
...
ii rsync 2.6.9-3cnl.1 Modified by CN 2007-09-02 to include md5sums.
```

ПРИМЕЧАНИЕ

Более подробную информацию о формировании DEB-архивов вы можете найти в руководстве Debian Binary Package Building HOWTO, размещенном по адресу http://tldp.org/HOWTO/Debian-Binary-Package-Building-HOWTO. MAN-страница, посвященная dpkg-deb, также содержит информацию о формировании DEB-архивов.

Резюме

Программное обеспечение Ubuntu и других дистрибутивов на базе Debian упаковывается в архивы формата DEB. По умолчанию для установки Ubuntu используется установщик Ubiquity. Boot-меню позволяет загружать все необходимые для установки приложения, устанавливать систему, а также запускать Ubuntu с компакт-диска. Для загрузки и установки дополнительного программного обеспечения из интернет-репозиториев программного обеспечения вы можете использовать программы aptitude и APT. Для установки программных пакетов, хранящихся на жестком диске, а также индивидуально сформированных пакетов Debian вы можете пользоваться утилитой dpkg. И APT, и aptitude, и dpkg предоставляют возможность получать информацию о программном обеспечении. Используя приложения debsums и md5sum, вы можете проверять установленные пакеты.

3 Использование командного процессора

Использование интерпретатора командного языка (обычно называется просто командным процессором, или консолью) восходит к первым операционным системам UNIX. Помимо очевидного использования командного процессора для выполнения команд, он имеет множество встроенных параметров, таких как переменные окружения, псевдонимы, а также множество функций, предназначенных для программирования. Хотя наиболее часто используемый командный процессор, применяемый в системах Linux, называется Bourne Again Shell (bash), существуют и другие командные консоли (например, sh, csh, ksh, tcsh и др.). Во многих случаях эти командные консоли (например, sh) являются символьными ссылками на другие консоли, такие как bash. В Ubuntu Linux sh является символьной ссылкой на `/bin/dash`. Командный процессор sh является важной частью системы, поскольку в большинстве сценариев командного процессора он определяется как консоль запуска сценариев. Что касается интерактивного использования, то в качестве командного процессора по умолчанию используется bash.

В данной главе приведен материал, призванный помочь вам научиться работать в командных консолях Linux вообще и консоли bash в частности.

Окна терминала и доступ к командному процессору

Самым простым способом запуска командного процессора из графического интерфейса Linux является использование окна терминала. Как правило, получить доступ к виртуальному терминалу, а также открыть командную консоль можно и из графического интерфейса, однако если у вас не установлена графическая оболочка, а есть только текстовая, то командная консоль запускается сразу после входа в систему.

Использование окон терминала

Чтобы открыть окно терминала из графической оболочки GNOME (графическая оболочка Ubuntu по умолчанию), откройте меню **Applications ▸ Accessories ▸ Terminal** (Приложения ▸ Инструменты ▸ Терминал). В результате откроется окно **gnome-terminal**,

отображающее командную строку консоли bash. На рис. 3.1 показан пример окна gnome-terminal.

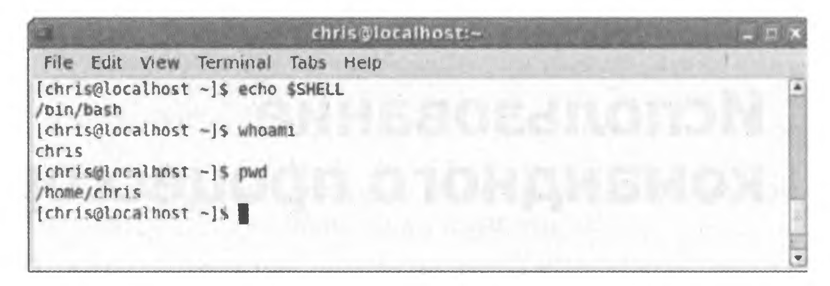


Рис. 3.1. Ввод команд в командный процессор через окно gnome-terminal

Команды, показанные на рис. 3.1, свидетельствуют, что *текущей командной консолью* является bash (/bin/bash), *текущим пользователем* — пользователь настольного компьютера, запустивший окно (chris), а *текущей папкой* — личная папка пользователя (/home/chris). Имя пользователя (chris) и имя хост-системы (localhost) отображаются и в строке заголовка.

Окно gnome-terminal не только предоставляет доступ к командному процессору, но и дает возможность управлять командными консолями. Например, чтобы **открыть другую консоль или поменять вкладку**, откройте меню File ▸ Open Tab (Файл ▸ Открыть вкладку); чтобы **открыть новое окно терминала**, откройте меню File ▸ Open Terminal (Файл ▸ Открыть терминал); а чтобы указать новый заголовок окна, откройте меню Terminal ▸ Set Title (Терминал ▸ Задать заголовок).

Кроме того, для работы в окне терминала можно использовать сочетания клавиш. Нажав сочетание клавиш Shift+Ctrl+T, можно **открыть консоль на новой вкладке**; нажатием сочетания клавиш Shift+Ctrl+N **открывается новое окно терминала**; с помощью сочетания клавиш Shift+Ctrl+W **закрывается вкладка**, а Shift+Ctrl+Q — **окно терминала**. С помощью сочетания клавиш Shift+Ctrl+C можно скопировать выделенный текст, а чтобы **вставить скопированный текст в то же или другое окно**, нужно нажать сочетание клавиш Shift+Ctrl+V или среднюю кнопку мыши.

ПРИМЕЧАНИЕ

В большинстве приложений, таких как, например, текстовый редактор OpenOffice.org, копирование осуществляется нажатием сочетания клавиш Ctrl+C, а не Shift+Ctrl+C, а вставка — сочетания клавиш Ctrl+V, а не Shif+Ctrl+V. Поскольку сочетание клавиш Ctrl+C в окне командного процессора носит специальное назначение (этим сочетанием клавиш обычно закрывается программа), в окне gnome-terminal функции графического интерфейса реализуются с помощью добавления клавиши Shift.

Среди других горячих клавиш, используемых при управлении окнами терминала, можно назвать клавишу F11, используемую для перехода в *полноэкранный режим*. Для увеличения размера отображаемого текста используйте сочетание клавиш Ctrl+Shift++, а для уменьшения его размера — Ctrl+–. **Переключение между вкладками** осуществляется с помощью сочетаний клавиш Ctrl+Page Up и Ctrl+Page Down (предыдущая и следующая вкладка соответственно) или Alt+1,

Alt+2, Alt+3 и т. д. для перехода соответственно на первую, вторую, третью и т. д. вкладку. Для редактирования командной консоли нажмите сочетание Ctrl+D, после чего закроется текущая вкладка или окно терминала (если это последняя вкладка).

Окно gnome-terminal также поддерживает профили (откройте меню Edit ▸ Current Profile (Редактировать ▸ Текущий профиль)). Некоторые настройки профилей определяют оформление окна (разрешают использование *четкого текста*, *мерцания курсора*, *звуков терминала*, *цветов*, *рисунков* и *прозрачности*). Другие установки являются функциональными. Например, терминал сохраняет по умолчанию 500 полос прокрутки (318 Кбайт). Некоторые хотят, чтобы сохранялось больше, и согласны выделить для этого больше памяти.

Если вы используете ручной запуск окна gnome-terminal, то можете добавлять параметры, например:

```
$ gnome-terminal -x alsamixer      Запускает терминал с отображенным alsamixer
$ gnome-terminal --tab --tab -tab  Запускает терминал с тремя открытыми вкладками
$ gnome-terminal --geometry 80x20  Запускает терминал размером 80 символов
                                   на 20 строк
$ gnome-terminal --zoom=2          Запускает терминал с более крупным шрифтом
```

Помимо окна gnome-terminal, существует множество других окон терминала, которые вы можете использовать. Вот несколько примеров: xterm (основной эмулятор терминала, используемый системой X Window), aterm (эмулятор терминала, созданный по аналогии с эмулятором Afterstep XVT VT102) и konsole (эмулятор терминала, поставляемый вместе с графической оболочкой KDE). Проект Enlightenment desktop предоставляет терминал eterm, который включает в себя такие элементы, как журнал сообщений на фоне экрана.

Работа с виртуальными терминалами

Когда Ubuntu загружается в многопользовательском режиме (уровень 2, 3 или 5), создается шесть виртуальных консолей (от tty1 до tty6) с текстовыми учетными записями. Если используется Рабочий стол системы X Window System, то X, возможно, функционирует в виртуальной консоли 7. Если это не так, то, скорее всего, перед вами виртуальная консоль 1.

Из X с помощью нажатия сочетаний клавиш Ctrl+Alt+F1, Ctrl+Alt+F2 и т. д. до 6 осуществляется **переключение на другую виртуальную консоль**. Находясь в текстовой виртуальной консоли, вы можете переключаться между ними с помощью сочетаний клавиш Alt+F1, Alt+F2 и т. д. Для возврата в GUI X нажмите Alt+F7. Каждая консоль для входа в систему использует различные учетные записи пользователей. Переключение на другую консоль не прерывает выполнения процессов остальных консолей. Если вы переключитесь на какой-либо виртуальный терминал, то увидите командную строку учетной записи пользователя примерно следующего содержания:

```
Ubuntu 7.04 localhost tty2
localhost login:
```

Каждым виртуальным терминалом управляют отдельные процессы getty. Чтобы увидеть, что представляют из себя процессы getty, перед входом в любой виртуальный терминал введите следующую команду:

```
$ ps awx | grep -v grep | grep getty
 4366 tty4      Ss+         0:00 /sbin/getty 38400 tty4
 4367 tty5      Ss+         0:00 /sbin/getty 38400 tty5
 4372 tty2      Ss+         0:00 /sbin/getty 38400 tty2
 4373 tty3      Ss+         0:00 /sbin/getty 38400 tty3
 4374 tty1      Ss+         0:00 /sbin/getty 38400 tty1
 4375 tty6      Ss+         0:00 /sbin/getty 38400 tty6
```

После входа в первую консоль getty идентифицирует учетную запись, а затем запускает командный процессор bash:

```
$ ps awx | grep -v grep | grep tty
 4366 tty4      Ss+         0:00 /sbin/getty 38400 tty4
 4367 tty5      Ss+         0:00 /sbin/getty 38400 tty5
 4372 tty2      Ss          0:00 /bin/login --
 4373 tty3      Ss+         0:00 /sbin/getty 38400 tty3
 4374 tty1      Ss+         0:00 /sbin/getty 38400 tty1
 4375 tty6      Ss+         0:00 /sbin/getty 38400 tty6
 7214 tty2      S+          0:00 -bash
```

Настройки виртуальных консолей хранятся в папке /etc/event.d. Для каждой виртуальной консоли предусмотрен сценарий, например tty1 для консоли **tty1**, tty2 для консоли **tty2** и т. д.

ПРИМЕЧАНИЕ

В большинстве версий Linux настройки консолей хранятся в файле /etc/inittab. Демон[1] в качестве своего конфигурационного файла по умолчанию использует /etc/inittab. В Ubuntu Linux же программа по умолчанию заменяется новой программой, называемой upstart, которая для хранения своих конфигурационных файлов использует папку /etc/event.d.

Работа в командном процессоре

Когда вы открываете командный процессор (неважно, через текстовую учетную запись или через окно терминала), в зависимости от пользователя, открывшего командную консоль, устанавливается ее окружение. Установки командной консоли bash для пользователей всех консолей хранятся в нескольких файлах. Чтобы изменить настройки системы, вы можете создать свои собственные версии этих файлов. Существует два типа файлов, содержащих эти установки: файлы запуска и файлы инициализации.

Консоль bash запускает загрузочные файлы для всех исходных командных процессоров. Эти файлы определяют установки, которые применяются во всей

[1] Скрытая от пользователя служебная программа, вызываемая при выполнении какой-либо функции.

учетной записи. Консоль bash запускает файлы инициализации, чтобы консоли работали интерактивно, то есть не запуская сценарий командного процессора.

Bash ищет файлы запуска в папке /etc/profile (общесистемные), а индивидуальные настройки — в некоторых файлах, начинающихся с точки в основной папке пользователя (при ее наличии): .bash_profile, .bash_login и .profile.

ПРИМЕЧАНИЕ

В других версиях Linux общесистемные файлы хранятся в папках /etc/profile и /etc/profile.d/.

Файлы же инициализации bash ищет в папке /etc/bash.bashrc (общесистемные), а индивидуальные настройки — в файле с расширением BASHRC вашей основной папки.

ПРИМЕЧАНИЕ

В других версиях Linux общесистемные файлы хранятся в папке /etc/bashrc.

Если командный процессор закрыт, то все команды выполняются в пользовательском файле ~/.bash_logout. Изменение настроек этих файлов изменяет пользовательские настройки консоли, но не затрагивает уже запущенные консоли (другие консоли используют собственные конфигурационные файлы).

Существует множество способов, позволяющих просматривать и изменять среду вашей консоли. Одним из основных является смена пользователя; в частности, изменение простой учетной записи на учетную запись суперпользователя (см. следующий раздел).

Журнал bash

Консоль, называемая Bourne Again Shell (bash), является командным процессором, используемым по умолчанию на большинстве современных операционных систем Linux и немного в других операционных системах, например Mac OS X. В консоль bash, как и в другие командные консоли, встроен элемент, называемый «журналом», который позволяет просматривать, изменять и отменять команды, выполненные ранее. Эта функция может оказаться очень полезной, поскольку многие команды в Linux являются длинными и сложными.

При запуске bash считывается и загружается в память файл ~/.bash_history. Значение этого файла по умолчанию устанавливается в $HISTFILE.

ПРИМЕЧАНИЕ

Для получения более подробной информации о работе с такими переменными окружения командного процессора, как $HISTFILE, ознакомьтесь с разд. «Использование переменных среды».

Во время сеанса bash команды добавляются в журнал, находящийся в памяти. После выхода из консоли bash данные из журнала переписываются в файл .bash_history. **Количество команд, сохраняемых в журнале в течение сеанса bash**, задается в файле $HISTSIZE, а **количество команд, хранящихся в файле журнала в данный момент**, определяется файлом $HISTFILESIZE:

```
$ echo $HISTFILE $HISTSIZE $HISTFILESIZE
/home/fcaen/.bash_history 500 500
```

Для **просмотра всего журнала** выполните команду history. Для **просмотра же определенного количества ранее выполненных и записанных в журнал** команд добавьте после команды history необходимое число. Команда, приведенная ниже, отображает предыдущие пять команд, хранящихся в журнале:

```
$ history 5
975 mkdir extras
976 mv *doc extras/
977 ls -CF
978 vi house.txt
979 history
```

Чтобы **перемещаться среди команд**, записанных в журнал, используйте клавиши управления курсором ↑ и ↓. Когда необходимая команда будет найдена, вы сможете использовать клавиатуру для **редактирования текущей команды**: клавиши ←, →, Delete, Backspace и т. д. Ниже показаны некоторые другие способы восстановления и запуска команд из журнала bash:

```
$ !!             Выполняет предыдущую команду
$ !997           Выполняет команду, записанную в журнале под номером 997
ls -CF
$ !997 *doc      Добавляет расширение DOC к команде из журнала
ls -CF *doc
$ !?CF?          Выполняет предыдущую команду, содержащую строку CF
ls -CF *doc
$ !ls            Выполняет предыдущую команду ls
ls -CF *doc
$ !ls:s/CF/l     Выполняет предыдущую команду, изменяя CF на l
ls -l *doc
```

Еще одним способом **редактирования журнала команд** является применение команды fc, с помощью которой, используя редактор vi, можно открыть выбранную из журнала команду. Измененная команда запускается сразу после закрытия окна редактора. Перейти в другой редактор можно, задав переменную FCEDIT (например, FCEDIT=gedit) или выполнив через командную строку команду fc:

```
$ fc 978                   Позволяет отредактировать команду
                           под номером 978, а затем выполнить ее
$ fc                       Позволяет отредактировать предыдущую команду,
                           а затем выполнить ее
$ fc -e /usr/bin/nano 989  Открывает редактор nano для редактирования
                           команды под номером 989
```

Для **поиска нужной строки в журнале** используется сочетание клавиш Ctrl+R. Например, если нажать Ctrl+R, а затем ввести строку ss, то будет выведена следующая информация:

```
# <Ctrl+R>
(reverse-i-search)`ss': sudo /usr/bin/less /var/log/messages
```

Чтобы **выполнить реверсивный поиск в журнале** на наличие строк, содержащих строку ss, повторно нажмите сочетание клавиш Ctrl+R.

ПРИМЕЧАНИЕ

По умолчанию для редактирования команд журнала bash используется редактор emacs, однако если вы предпочитаете редактор vi, то можете использовать и его. Чтобы установить vi в качестве редактора команд журнала, воспользуйтесь командой set: введите set -o vi.

Функция дополнения командной строки

Для дополнения командной строки различной информацией используется клавиша Tab. Ниже приведено несколько примеров, демонстрирующих быстрый ввод команд, **осуществляемый частичным вводом команды, а затем нажатием клавиши Tab**:

```
$ tracer <Tab>          Дополняет команду до команды traceroute
$ Cd /Home/Ch <Tab>     Дополняет путь до записи /home/chris
$ cd ~jo <Tab>          Дополняет название основной папки пользователя
                        до /home/john
$ echo $PA <Tab>        Дополняет переменную окружения до $PATH
$ ping <Alt+@> <Tab>    Дополняет хост-машины: показывает хост-машины
                        из /etc/hosts
@davinci.example.com    @ritchie.example.com @thompson.example.com
@localhost              @zooey
```

Переназначение stdin и stdout

Команда, введенная в командную консоль, выполняется в диалоговом режиме. В результате образуются два варианта потока: stdout (если команда выполнена нормально) и stderr (если при выполнении команды произошла ошибка). Ниже показано, что при поиске несуществующего файла или папки с именем /tmpp на строку stderr выдается сообщение об ошибке, а при поиске /tmp (который успешен) данные, выводимые согласно запросу, отображаются на строке stdout:

```
$ ls /tmp /tmpp
ls: /tmpp: No such file or directory
/tmp/:
gconfd-fcaen keyring-b41WuB keyring-ItEWbz mapping-fcaen orbit-fcaen
```

По умолчанию на экран выводятся все данные. Чтобы **направить выводимые данные в файл**, используйте знак >. В частности, используя знак >, вы можете направить в файл стандартный поток вывода (stsndsrt output stream — stdout), а используя сочетание 2>, — стандартный поток ошибки (standart error stream — stderr):

```
$ ls /tmp /tmmp > output.txt
ls: /tmpp: No such file or directory
$ ls /tmp /tmmp 2> errors.txt
/tmp/:
gconfd-fcaen keyring-b41WuB keyring-ItEWbz mapping-fcaen orbit-fcaen
$ ls /tmp /tmmp 2> errors.txt > output.txt
$ ls /tmp /tmmp > everything.txt 2>&1
```

В первом примере stdout перенаправлен в файл output.txt, а stderr выводится на экран. Во втором же примере stderr направлен в файл errors.txt, а stdout выводится на экран. В третьем примере объединены первые два примера. В последнем примере оба потока направлены в файл everything.txt. Чтобы **дописать в файл информацию**, а не переписывать ее, используйте сразу два знака >:

```
$ ls /tmp >> output.txt
```

Если вам вообще не нужно видеть поток вывода, то можете просто **направить его в специальный файл-«битоприемник»** (/dev/null):

```
$ ls /tmp 2> /dev/null
```

СОВЕТ

Другим случаем, при котором может понадобиться перенаправить поток stderr, является ситуация, когда вы работаете с crontab. Вы можете перенаправить stderr через электронное сообщение, которое будет направлено владельцу. В этом случае пользователю не будет возвращаться никаких сообщений об ошибке.

Вы можете не только направлять поток стандартного вывода команды, но и **задавать канал стандартного ввода команды**. Например, следующая команда направляет файл /etc/hosts пользователю локальной системы chris в виде электронного сообщения:

```
$ mail chris < /etc/hosts
```

Используя каналы, вы можете **перенаправлять вывод данных с одного процесса на другой**, а не просто в другие файлы. В следующем примере данные из команды ls выводятся в команду sort для упорядочивания выводимых данных:

```
$ ls /tmp | sort
```

В следующем примере одновременно **присутствуют и канал, и направление** (поток stdout команды ls упорядочивается, а вывод stderr направляется в /dev/null):

```
$ ls /tmp/ /tmmp 2> /dev/null | sort
```

Каналы можно использовать для нескольких задач:

```
$ dpkg-query -l | grep -i sql | wc -l
$ ps auwx | grep firefox
$ ps auwx | less
$ whereis -m bash | awk '{print $2}'
```

В предыдущем примере первая строка отображает все установленные программные пакеты, выделяет те из них, которые содержат sql, и подсчитывает количество пропущенных строк (фактически подсчитывая количество пакетов, содержащих в своем имени sql). Вторая команда отображает процессы Firefox, взятые из длинного списка процессов (если браузер Firefox запущен), а также любые процессы, командная строка которых содержит ссылку на слово firefox. Третья команда позволяет пролистать весь список процессов. Последняя строка отображает слово bash, за ним идет путь к странице MAN-справочника, посвященной bash, а затем

команда, отображающая только путь к данной странице man (второй элемент на строке).

Используя обратные одинарные кавычки, вы можете **сначала выполнить часть командной строки, а затем вывести на оставшуюся часть командной строки данные этой команды**. Например:

```
$ dpkg-query -S `which ps`
$ ls -l `which bash`
```

Первая строка в предыдущем примере находит полный путь команды ps, а также программный пакет, содержащий эту команду. Вторая команда находит полный путь к команде bash и создает длинный список (ls -l) этой команды.

Более сложным и эффективным способом **выделить выводимые данные одной команды и применить их в качестве параметров** для другой является использование команды xargs:

```
$ ls /bin/b* | xargs dpkg-query -S
```

Чтобы показать, что команда xargs сейчас будет запущена, выполните следующую команду:

```
$ ls /bin/b* | xargs -t dpkg-query -S
dpkg-query -S /bin/bash /bin/bunzip2 /bin/bzcat /bin/bzcmp /bin/bzdiff
/bin/bzegrep /bin/bzexe /bin/bzfgrep /bin/bzgrep /bin/bzip2 /bin/bzip2recover
/bin/bzless /bin/bzmore
bash: /bin/bash
bzip2: /bin/bunzip2
bzip2: /bin/bzcat
bzip2: /bin/bzcmp
bzip2: /bin/bzdiff
bzip2: /bin/bzegrep
bzip2: /bin/bzexe
bzip2: /bin/bzfgrep
bzip2: /bin/bzgrep
bzip2: /bin/bzip2
bzip2: /bin/bzip2recover
bzip2: /bin/bzless
bzip2: /bin/bzmore
```

В этом примере все данные, выводимые ls, перенаправляются в команду dpkg-query -S. Если в xargs применить параметр -t, то выходные данные команды будут выведены еще до того, как она будет выполнена. Теперь воспользуемся xargs, чтобы перенаправить выходные данные ls для каждой отдельной команды dpkg-query. Здесь знак {} определен как поле для подстановки строки:

```
$ ls /bin/b* | xargs -t -I{} dpkg-query -S {}
dpkg-query -S /bin/bash
bash: /bin/bash
dpkg-query -S /bin/bunzip2
bzip2: /bin/bunzip2
dpkg-query -S /bin/bzcat
bzip2: /bin/bzcat
```

```
dpkg-query -S /bin/bzcmp
bzip2: /bin/bzcmp
dpkg-query -S /bin/bzdiff
bzip2: /bin/bzdiff
dpkg-query -S /bin/bzegrep
bzip2: /bin/bzegrep
dpkg-query -S /bin/bzexe
bzip2: /bin/bzexe
dpkg-query -S /bin/bzfgrep
bzip2: /bin/bzfgrep
dpkg-query -S /bin/bzgrep
bzip2: /bin/bzgrep
dpkg-query -S /bin/bzip2
bzip2: /bin/bzip2
dpkg-query -S /bin/bzip2recover
bzip2: /bin/bzip2recover
dpkg-query -S /bin/bzless
bzip2: /bin/bzless
dpkg-query -S /bin/bzmore
bzip2: /bin/bzmore
```

Как вы можете видеть из выводимых данных, для каждого параметра, касающегося `ls`, запускаются отдельные команды `dpkg-query -S`.

Алиасы

Чтобы **установить и просмотреть алиасы**, необходимо воспользоваться командой `alias`. Некоторые алиасы уже установлены в общесистемных или пользовательских файлах инициализации консоли, о которых говорилось ранее. Следующий пример показывает, как **отобразить уже установленные алиасы**:

```
$ alias
alias cp='cp -i'
alias ls='ls --color=auto'
alias mv='mv -i'
alias rm='rm -i'
```

Отметим, что некоторые алиасы установлены просто в качестве способа добавления параметра в изначальные характеристики команды (например, `mv -i`, применяемый, если необходимо переместить файл).

ПРИМЕЧАНИЕ

Ubuntu Linux работает только с алиасами, определяемыми ls, которые при просмотре файлов выделяют выходные данные цветом. Другие алиасы представляют собой примеры полезных команд, которыми вы можете захотеть воспользоваться, особенно после того, как они помогут вам предотвратить случайное удаление файлов.

Определить собственные алиасы для текущего сеанса bash можно следующим образом:

```
$ alias la='ls -la'
```

Добавьте эту строку в файл ~/.bashrc для определения собственных алиасов в каждом новом сеансе bash. Используя команду unalias, можно **удалить алиас из текущей сессии bash**:

```
$ unalias la        Удаляет предыдущий созданный алиас команды la
$ unalias -a        Удаляет все созданные алиасы
```

Наблюдение за командами

Если вам необходимо следить за командами с изменяющимися выводимыми данными, используйте команду watch. Например, чтобы следить за средней загрузкой, выполните следующую команду:

```
$ watch 'cat /proc/loadavg'
```

Каждые две секунды watch будет запускать команду cat. Для завершения выполнения команды нажмите сочетание клавиш Ctrl+C. Чтобы увеличить частоту обновления до 10 секунд, выполните следующую команду:

```
$ watch -n 10 'ls -l'
```

Чтобы выделить разницу между обновлениями на экране, выполните:

```
$ watch -d 'ls -l'
```

Чтобы завершить выполнение команды watch, нажмите сочетание клавиш Ctrl+C. Стоит отметить, что для выполнения выделения необходимо, чтобы файлы изменялись.

Наблюдение за файлами

Для наблюдения за размером файлов также может быть использована команда watch. Например, чтобы следить за размером большого ISO-файла mydownload.iso по мере его загрузки, воспользуйтесь следующей командой:

```
$ watch 'ls -l mydownload.iso'
```

Чтобы наблюдать за содержимым файла с открытым текстом, размер которого увеличивается с течением времени, воспользуйтесь командой tail. Например, вы можете просмотреть сообщения, добавленные в файл /var/log/message, с помощью команды:

```
$ sudo tail -f /var/log/messages
```

Чтобы завершить выполнение команды tail, нажмите Ctrl+C.

Получение прав суперпользователя

Когда вы запускаете командный процессор, вы можете выполнять команды и получать доступ к файлам и папкам на основе ID пользователей и групп, а также прав доступа к этим компонентам. Доступ ко многим системным ресурсам является

ограниченным и открыт только для *пользователя root*, также называемого *суперпользователем*.

Существует три основных способа получения прав суперпользователя:

- войти в систему как суперпользователь;
- временно получить права суперпользователя, воспользовавшись командой su;
- выполнить одну команду на правах суперпользователя, воспользовавшись командой sudo.

В большинстве случаев в систему не входят от имени суперпользователя, так как это повышает вероятность случайно внести нежелательные изменения в систему. Большинство пользователей Linux используют либо команду su (для перехода из обычной учетной записи в учетную запись суперпользователя), либо команду sudo (для выполнения одной команды с привилегиями суперпользователя).

Ubuntu Linux предоставляет пользователям возможность выполнять команду sudo. Поэтому, чтобы выполнить команду, связанную с администрированием (например, команду useradd, служащую для добавления нового пользователя), можно сначала ввести команду sudo:

```
$ sudo useradd -m joe     На правах суперпользователя добавляет нового
                          пользователя по имени joe
```

По умолчанию в операционной системе Ubuntu существуют ограничения, не позволяющие создать учетную запись суперпользователя. Поэтому в Ubuntu не предусмотрена и возможность выполнения команды su, которая обычно используется в других операционных системах Linux и которая дает возможность переключаться на учетную запись суперпользователя.

Если вам нужно выполнить некоторое количество команд, используя права суперпользователя, вы можете выполнить команду, которая позволяет запускать командный процессор от имени пользователя root:

```
$ sudo bash            Запускает командную консоль от имени суперпользователя
#
```

Если же вам нужно создать пароль для учетной записи суперпользователя (что позволит вам как входить в систему через учетную запись суперпользователя, так и использовать команду su, чтобы временно им становиться), воспользуйтесь командой sudo:

```
$ sudo passwd root     Устанавливает пароль учетной записи суперпользователя
```

Впрочем, большинство пользователей Ubuntu просто используют sudo и никогда не задают root-пароль.

Использование команды su

Если в какой-то момент вы все же решите создать пароль для учетной записи суперпользователя с открытой командной строкой, то, чтобы **стать суперпользова-**

телем, воспользуйтесь командой su. Кроме того, вы можете использовать команду su для переключения на учетную запись другого пользователя, не являющегося суперпользователем. В следующих примерах описываются принципы работы команды su.

Простое выполнение команды su, продемонстрированное ниже, не предоставляет доступа к исходному командному процессору со средой суперпользователя:

```
$ su
Password:*****
# echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games
/usr/kerberos/sbin:/usr/kerberos/bin:/usr/local/bin:/usr/bin:/bin:/usr/X11R6/bin
:/home/fcaen/bin
```

После выполнения команды su пользователь все еще обладает переменной PATH пользователя fcaen. Чтобы **получить доступ к среде суперпользователя**, используйте команду su с дефисом (-):

```
# exit
$ su -
Password: *****
# echo $PATH
/usr/kerberos/sbin:/usr/kerberos/bin:/usr/local/sbin:/usr/local/bin:/sbin:/bin:/
usr/sbin:/usr/bin:/root/bin
```

В большинстве случаев для получения прав суперпользователя достаточно использовать su -, если нет острой необходимости воспользоваться другим способом. Если не обозначен ни один пользователь, su устанавливает суперпользователя по умолчанию. Однако команду su можно также применять, чтобы **переключаться на других пользователей**:

```
$ su - cnegus
```

Команда su также может быть использована **для выполнения отдельной команды от лица конкретного пользователя**:

```
$ su -c whoami
Password: ******
root
# su -c 'less /var/log/messages'
```

Несмотря на то что во втором примере вход в систему изначально был осуществлен от лица обычного пользователя, выполнение команды whoami с параметром su -c говорит о том, что вы являетесь суперпользователем. В первом же примере с обеих сторон командной строки less необходимо установить кавычки, которые служат для определения строки /var/log/messages как параметра команды less. Как уже было сказано, команда whoami может быть полезна для **определения используемой для выполнения текущей команды учетной записи**:

```
$ whoami
fcaen
```

Распределение прав с помощью команды sudo

Команда sudo позволяет очень четко разделять права доступа пользователей, не имеющих доступа к учетной записи суперпользователя. Команда sudo является прекрасным инструментом для предоставления расширенных привилегий в случаях, когда в системе используется несколько учетных записей пользователей, а также для сбора информации о том, каким образом пользователи используют предоставленные им привилегии.

Если не задано иное, sudo активизирует учетную запись суперпользователя. В Ubuntu Linux для выполнения команд, требующих обладания привилегиями суперпользователя, применяется команда sudo, а не su.

Настройки команды sudo хранятся в файле /etc/sudoers.

ВНИМАНИЕ

Никогда не редактируйте этот файл в обычном текстовом редакторе — вместо этого всегда используйте команду visudo.

Доступ к файлу /etc/sudoers ограничен, поэтому для его редактирования необходимо использовать команду sudo:

```
$ sudo visudo
```

Команда visudo запускает редактор (по умолчанию nano, про который рассказывалось ранее).

Если вы ознакомитесь с содержанием файла sudoers, поставляемого с вашим дистрибутивом, то увидите пустые разделы, разделенные примечаниями, и один активный оператор:

```
root     ALL=(ALL)   ALL
```

Это значит, что пользователь **root** может выполнить любую команду от имени любого пользователя на любой хост-машине. Чтобы все пользователи, входящие в группу администратора, могли получить привилегии суперпользователя, в Ubuntu Linux добавлена следующая срока:

```
%admin   ALL=(ALL)   ALL
```

После того как вы установите Ubuntu Linux, учетная запись пользователя, созданная вами, автоматически добавится к этой группе. Чтобы дополнительным пользователям были доступны root-привилегии, добавьте в файл следующую строку, изменив значение первого поля на имя учетной записи пользователя вашей системы:

```
Fcaen    ALL= /usr/bin/less /var/log/messages
```

ПРИМЕЧАНИЕ

Предыдущая настройка позволяет пользователю выполнять команду less с привилегиями суперпользователя. Это действие ставит под угрозу безопасность системы, поскольку команда less позволяет пользователю посредством изучения других системных файлов получить больше информации о системе.

Теперь пользователь **fcaen** (или любой другой добавленный вами пользователь) может выполнить следующую команду:

```
$ sudo /usr/bin/less /var/log/messages
Password:
```

После того как пользователь **fcaen** введет свой пароль, он сможет просматривать файл /var/log/messages. Одновременно с этим будет установлена и временная метка, что позволит этому пользователю в следующие пять минут (время по умолчанию) вводить команды в командную строку и выполнять их без необходимости указывать пароль.

Однако обычно приходится добавлять определенных пользователей в группу **admin**, а не создавать индивидуальные записи в файле /etc/sudoers.

Каждое использование sudo регистрируется в файле /var/log/secure:

```
Feb 24 21:58:57 localhost sudo: fcaen : TTY=pts/3 ; PWD=/home/fcaen ;
USER=root ; COMMAND=/usr/bin/less /var/log/messages
```

Далее добавьте следующую строку в /etc/sudoers:

```
Fcaen     serverl=(chris)    /bin/ls /home/chris
```

Теперь пользователь **fcaen** может выполнить следующую команду:

```
$ sudo -u chris /bin/ls /home/chris
```

Команда sudo, приведенная выше, будет выполнена от имени пользователя **chris** и будет работать только на хост-сервере 1. В некоторых организациях управление файлом /etc/sudoers осуществляется централизованно и он используется на всех хост-машинах, поэтому может оказаться полезным на отдельных машинах точно определить права доступа к sudo.

Команда sudo также позволяет задавать алиасы или заранее заданные группы пользователей, команд и хост-машин. Поищите соответствующие примеры в файле /etc/sudoers своей операционной системы Linux.

Переменные среды

Небольшие фрагменты информации, которые могут быть полезны для конфигурации командной консоли, находятся в так называемых *переменных среды*. Обычно имена переменных среды всегда пишутся с заглавных букв (хотя у этого правила есть исключения). Если вы используете командный процессор bash, то исходя из различных начальных сценариев bash, описанных ранее, некоторые переменные среды будут уже установлены.

Чтобы **отобразить все переменные среды**, уже установленные для вашей консоли, в алфавитном порядке, выполните следующую команду:

```
$ set | less
BASH=/bin/bash COLORS=/etc/DIR_COLORS.xterm COLUMNS=118
DISPLAY=:0.0
```

```
HOME=/home/fcaen
HOSTNAME=einstein
...
```

Приведенные выходные данные составляют лишь небольшую часть переменных, которые будут отображены у вас. Команда set также отображает и функции. Команда env отображает только переменные среды.

Кроме того, вы можете **устанавливать или сбрасывать переменные самостоятельно**. Например, чтобы присвоить переменной ABC значение 123 (а затем отобразить содержимое ABC), выполните следующую команду:

```
$ ABC=123
$ echo $ABC
123
```

Переменная ABC существует только в той командной консоли, в которой она была создана. Если вы выполните команду из другой консоли (ls, cat, firefox и т. д.), то этому новому процессу созданная переменная не будет доступна. Запустите новый процесс bash и проверьте это:

```
$ bash
$ echo $ABC
$
```

Кроме того, **экспортировав переменные, вы можете сделать их частью среды, которая передается дочерними процессами**:

```
$ export ABC=123
$ bash
$ echo $ABC
123
```

Вы также можете **объединить строку и существующую переменную**:

```
$ export PATH=$PATH:/home/fcaen
```

Чтобы **отобразить все переменные окружения bash**, выполните следующую команду:

```
$ env
```

При создании собственных переменных среды избегайте использования уже задействованных системой имен. Список переменных среды командного процессора приведен в Приложении 2.

Создание простых сценариев для командного процессора

Сценарии командного процессора удобно использовать для автоматизации повторяющихся задач. В bash и другие командные консоли включены базовые компоненты, основанные на различных языках программирования, например: циклы, условия, операторы варианта и т. д. Главное различие между ними состоит в том, что есть только один тип переменных — строки.

Редактирование и запуск сценария

Сценарии командного процессора являются простыми текстовыми файлами. Вы можете создавать их, используя любой текстовый редактор (например, vi). Чтобы запустить сценарий, его файл должен быть исполняемым.

Например, если вы создали сценарий для командного процессора с именем файла myscript.sh, то **сделать его исполняемым** можно следующим образом:

```
$ chmod u+x myscript.sh
```

При этом первая строка вашего сценария для bash всегда должна выглядеть следующим образом:

```
#!/bin/bash
```

В данном случае комментарий начинается со знака #. Синтаксис #! применяется в качестве комментария для командных консолей, которые не распознают этот специальный синтаксис. Часть /bin/bash оповещает любой активный командный процессор (будь то bash или другая консоль), какую программу следует использовать для запуска сценария (поскольку раньше не все системы поставлялись с консолью bash, часто в роли команды для запуска сценария можно увидеть /bin/sh).

Как и с любой командой, помимо того, что любой сценарий командного процессора должен быть выполняемым, сценарий, создаваемый вами, должен при запуске являться частью переменной PATH или определяться по его полному или относительному пути. Другими словами, как только вы попробуете запустить сценарий, вы получите следующий результат:

```
$ myscript.sh
bash: myscript.sh: command not found      Команда не найдена
```

В этом примере папка, содержащая файл myscript.sh, не включена в переменную PATH. Для решения этой проблемы достаточно отредактировать путь, скопировать сценарий в папку переменной PATH или ввести полный или относительный путь к вашему сценарию. Все четыре примера приведены ниже:

```
$ mkdir ~/bin ; cp myscript.sh ~/bin/   PATH=$PATH:~/bin
$ cp myscript.sh /usr/local/bin
$ ./myscript.sh
$ /tmp/myscript.sh
```

Не следует ставить точку (.) в переменной окружения PATH с целью обозначить, что команда может быть выполнена из текущей папки, поскольку ее имя может совпасть с именем важной и широко используемой команды (например, ls или cat), что вызовет перезапись старых команд новыми, если у них будут одинаковые имена, и они окажутся в одной папке, а это может поставить под угрозу безопасность всей системы.

Добавление содержимого в сценарий

Хотя сценарии командного процессора могут представлять собой всего лишь последовательности команд, вы можете также по собственному усмотрению использовать их в языках программирования. Например, если изменять вводимые данные,

сценарий может возвращать различные результаты. В этом подразделе описывается, как можно использовать в сценариях командной консоли сочетания команд: например, операторы if/then, операторы выбора и циклы for/while.

Ниже приведена команда, назначающая строку abc переменной MYSTRING. Затем, чтобы узнать, равна ли переменная строке abc, она проверяет вводимые данные и далее действует исходя из результата проверки. Сам тест взят в квадратные скобки ([]):

```
MYSTRING=abc
if [ $MYSTRING = abc ] ; then
echo "The variable is abc"
fi
```

Чтобы **не проводить проверку**, воспользуйтесь сочетанием != вместо =:

```
if [ $MYSTRING != abc ] ; then
echo "$MYSTRING is not abc";
fi
```

Далее следуют примеры **тестирования на числа**:

```
MYNUMBER=1
if [ $MYNUMBER -eq 1 ] ; then echo "MYNUMBER equals 1"; fi
if [ $MYNUMBER -lt 2 ] ; then echo "MYNUMBER <2"; fi
if [ $MYNUMBER -le 1 ] ; then echo "MYNUMBER <=1"; fi
if [ $MYNUMBER -gt 0 ] ; then echo "MYNUMBER >0"; fi
if [ $MYNUMBER -ge 1 ] ; then echo "MYNUMBER >=1"; fi
```

Теперь взгляните на некоторые примеры **тестирования имен файлов**. В данном примере производится проверка существования указанного файла (-e) и его типа: обычный файл (-f) или папка (-d). Это осуществляется с помощью операторов if/then. Если совпадений нет, то используется оператор else.

```
filename="$HOME"
if [ -e $filename ] ; then echo "$filename exists"; fi
if [ -f "$filename" ] ; then
   echo "$filename is a regular file"
elif [ -d "$filename" ] ; then
   echo "$filename is a directory"
else
   echo "I have no idea what $filename is"
fi
```

В табл. 3.1 приведены примеры тестов, которые могут быть осуществлены с файлами, строками и переменными.

Таблица 3.1. Операторы для тестирования

Оператор	Описание теста
-a file	Проверяет наличие файла (то же, что -e)
-b file	Контролирует, какой файл является специальным блочным устройством
-c file	Проверяет, какой файл является специальным файлом устройства (например, устройства последовательной передачи данных)

Оператор	Описание теста
-d file	Контролирует, какой файл является каталогом
-e file	Проверяет наличие файла (то же, что -a)
-f file	Проверяет наличие файла и является ли он обычным (например, не является каталогом, сокетом, каналом, ссылкой или файлом устройства)
-g file	Обследует, какой файл обладает множеством битов set-group-id (назначение идентификатора группы)
-h file	Проверяет, какой файл имеет символьную ссылку (то же, что -L)
-k file	Контролирует, у какого файла есть бит закрепления в памяти
-L file	Проверяет, какой файл имеет символьную ссылку (то же, что -h)
-n string	Проверяет, длина какой строки превышает 0 байт
-O file	Контролирует, каким файлом вы владеете
-p file	Проверяет, какой файл является именованным каналом
-r file	Обследует, какой файл доступен вам для чтения
-s file	Проверяет, существует ли файл и превышает ли его размер 0 байт
-S file	Контролирует, существует ли файл и является ли он сокетом
-t fd	Проверяет, подключен ли дескриптор файла к терминалу
-u file	Поверяет, какой файл обладает множеством битов set-user-id (установка идентификатора пользователя)
-w file	Обследует, какой файл является перезаписываемым
-x file	Проверяет, какой файл является выполняемым
-z string	Контролирует, длина какой строки равна 0 байт
expr1 -a expr2	Проверяет, являются ли справедливыми и первое, и второе выражения
expr1 -o expr2	Проверяет, является ли справедливым одно из двух выражений
file1 -nt file2	Контролирует, является ли первый файл более новым по отношению ко второму (используется метка времени создания/модификации файла)
file1 -ot file2	Проверяет, является ли первый файл более старым по отношению ко второму (используется метка времени создания/модификации файла)
file1 -ef file2	Обследует, какой из двух файлов прикреплен к ссылке (жесткая ссылка или символьная)
var1 = var2	Проверяет, равна ли первая переменная второй
var1 -eq var2	Контролирует, равна ли первая переменная второй
var1 -ge var2	Поверяет, больше ли первая переменная второй или они равны
var1 -gt var2	Контролирует, больше ли первая переменная второй
var1 -le var2	Поверяет, меньше ли первая переменная второй или они равны
var1 -lt var2	Обследует, меньше ли первая переменная второй
var1 != var2 var1 -ne var2	Проверяет неравенство первой и второй переменных

Другим часто используемым структурным компонентом является команда case. Используя оператор case, можно проводить проверку переменных по различным критериям и исходя из результата действовать дальше. Так же, как и оператор switch, используемый в языках программирования, оператор case может использоваться вместо совмещенных операторов if.

```
case "$VAR" in
   string1)
         { action1 };;
   string2)
         { action2 };;
   *)
         { default action } ;;
esac
```

Примеры использования case вы можете найти в сценариях, применяемых при запуске системы и находящихся в папке /etc/init.d/. Каждый начальный сценарий выполняется на основе примененного ранее параметра (start, stop и т. д.), а его выбор осуществляется из большого количества вариантов.

ПРИМЕЧАНИЕ

Сценарии /etc/init.d выполняются посредством сценариев, хранящихся в папке /etc/event.d.

Командная консоль bash также позволяет выполнять стандартное построение циклов, что продемонстрировано ниже. В первом примере все значения переменной NUMBER (от 0 до 9 включительно) приведены в строке for:

```
for NUMBER in 0 1 2 3 4 5 6 7 8 9
do
   echo The number is $NUMBER
done
```

В следующих примерах выходные данные команды ls составляют переменные, в соответствии с которыми действует оператор for:

```
for FILE in `/bin/ls`; do echo $FILE; done
```

Чтобы не перечислять для оператора for все доступные значения, можно прирастить необходимое значение и **продолжать выполнение через цикл с проверкой условия, пока не будет установлено соответствие**. В следующем примере величина переменной VAR начинается со значения 0, и цикл с проверкой условия продолжает увеличивать его до тех пор, пока оно не возрастет до 3:

```
"VAR=0"
while [ $VAR -lt 3 ]; do
   echo $VAR
   VAR=$[$VAR+1]
done
```

Другим способом добиться аналогичного результата оператора цикла с условием продолжения является использование оператора until:

```
"VAR=0"
until [ $VAR -eq 3 ]; do echo $VAR; VAR=$[$VAR+1]; done
```

Если вы только начинаете осваивать программирование в командной консоли, ознакомьтесь с руководством Bash Guide for Beginners, доступным по адресу http://tldp.org/LDP/Bash-Beginners-Guide/html/index.html. Кроме того, вы можете воспользоваться справочным материалом, например man, для ознакомления с примерами по разработке качественных сценариев для командного процессора.

Резюме

Несмотря на усовершенствование графического интерфейса пользователя, командный процессор является одним из наиболее распространенных среди продвинутых пользователей методов работы в операционных системах Linux. Командная консоль Bourne Again Shell (bash) является наиболее распространенным командным процессором Linux. Она включает в себя множество полезных приложений, предназначенных для фиксирования и восстановления выполняемых команд (журнал), дополнения команд, установки алиасов и переназначения выводимых и вводимых данных. Вы также можете, используя простые техники написания сценариев для командного процессора, самостоятельно создавать эффективные команды.

4 Работа с файлами

В Linux любой элемент может быть отображен в виде файла, включая файлы данных, папки, устройства, именованные каналы, ссылки и другие типы файлов. В каждом файле содержится определенная информация, определяющая, кто может получить доступ к файлу и каким образом этот доступ может быть получен. В данной главе описано большое количество команд, позволяющих исследовать файлы и работать с ними.

Типы файлов

Папки и обычные файлы, несомненно, являются теми типами файлов, которые будут использоваться вами наиболее часто. Однако в Linux используются и некоторые другие типы файлов. Командная консоль предоставляет множество способов создавать, находить и просматривать различные типы файлов.

Файлы, предоставляющие доступ к устройствам компьютера, называются *файлами устройства*. Устройства делятся на символьные и блочные. Кроме того, существуют *жесткие ссылки* и *гибкие* (*символьные*), которые могут использоваться, чтобы сделать один и тот же файл доступным из разных мест. Реже обычными пользователями используются *именованные каналы* и *сокеты*, которые предоставляют точки входа для процессов, позволяющих обмениваться данными.

Обычные файлы

К обычным файлам относят файлы данных (документы, музыку, изображения и т. д.) и команды (двоичные и командные файлы). Определить тип файла позволяет команда `file`. Следующий пример демонстрирует способ перехода в папку с документацией, касающейся командного процессора bash, и использование команды `file` для отображения типов файлов в данной папке:

```
$ cd /usr/share/doc/
$ file doc-base/install-docs.html
doc-base/install-docs.html: XML 1.0 document text
$ file doc-base/copyright
doc-base/copyright: ASCII English text
$ file doc-base/doc-base.html
```

```
doc-base/doc-base.html/: directory
$ file doc/doc-base/changelog.gz
doc-base/changelog.gz: gzip compressed data, was "changelog", from Unix, last
modified: Thu Feb 22 07:29:26 2007, max compression
$ file shared-mime-info/shared-mime-info-spec.pdf
shared-mime-info/shared-mime-info-spec.pdf: PDF document, version 1.4
```

Команда file отображает файлы-документы различных форматов, находящиеся в папках, содержащих документацию по Ubuntu. Эта команда может определить, сжат ли текст, формат документа, в котором он может быть выведен на печать (PDF или PostScript), а также обычный ли это незашифрованный текст или он содержит метку HTML. Команда способна определить даже подпапки, что является достаточно неожиданным, поскольку они имеют дополнительные для них имена (doc-base.html).

Создавать обычные файлы можно посредством любого приложения, которое может сохранять данные. Если вы просто хотите **создать пустой файл, чтобы начать работу с него**, воспользуйтесь одним из следующих способов:

```
$ touch /tmp/newfile.txt     Создает пустой файл
$ > /tmp/newfile2.txt        Создает пустой файл
```

Определение полного описания файла является еще одним способом определить его тип:

```
$ ls -l /tmp/newfile2.txt    Отображает файл для определения его типа
-rw-r--r--      1 chris chris 0 Sep 5 14:19 newfile2
```

Знак дефиса (-) является первым символом десятисимвольной информации о правах доступа к файлу (-rw-r--r--), который свидетельствует о том, что данный файл является обычным (подробнее о правах доступа рассказывается в следующем разделе). Команды также являются обычными файлами, но сохраняются как исполняемые. Рассмотрим еще несколько примеров определения типа файла:

```
$ ls -l /usr/bin/apt-key
-rwxr-xr-x 1 root root 2230 2007-03-14 12:44 /usr/bin/apt-key
$ file /usr/bin/apt-key
/usr/bin/apt-key: Bourne shell script text executable
$ file /bin/ls
/bin/ls: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), for GNU/Linux
2.6.0, dynamically linked (uses shared libs), stripped
```

Как вы могли заметить, команда apropos выполняется через свойство x для владельца, группы и др. Выполнение команды file для файла apt-key позволяет увидеть, что он является сценарием командного процессора. Эта команда противопоставляется исполняемому двоичному коду, как, например, приведенная выше команда ls.

Каталоги

Каталог представляет собой место хранения файлов и подкаталогов. Они организуются в иерархическом порядке от корневого (/) до многочисленных подкаталогов,

отделяемых друг от друга символом /. При работе с каталогами через графические файловые менеджеры их принято называть *папками*.

Чтобы создать новый каталог для хранения данных, воспользуйтесь командой mkdir. Ниже приведены некоторые примеры использования данной команды, описывающие **различные пути создания каталогов**:

```
$ mkdir /tmp/new               Создает каталог new в /tmp
$ mkdir -p /tmp/a/b/c/new      Создает по мере необходимости каталог нижнего
                               уровня для new
$ mkdir -m 700 /tmp/new2       Создает каталог new2 с правами доступа drwx---
```

Первая команда mkdir создает новый каталог в уже существующем каталоге /tmp. Во втором примере каталоги создаются по мере необходимости (подкаталоги a, b и c), чтобы в результате был создан каталог new. В последнем примере, чтобы установить права доступа к каталогу, добавляется параметр -m.

Поскольку первым символом строки доступа к каталогу является буква d, данный **файл может быть определен как каталог**:

```
$ file /tmp/new
/tmp/new: directory
$ ls -l /tmp
...
drwxr-xr-x 2 ericfj ericfj 4096 2007-09-11 07:25 new
...
```

Говоря о каталогах, необходимо также отметить, что, если вы хотите, чтобы пользователи могли использовать свои каталоги в качестве рабочих, для них должны быть установлены исполняемые биты (x).

Символьные и жесткие ссылки

Чтобы избежать копирования файлов и каталогов в различные части файловой системы, можно использовать ссылки, позволяющие открывать доступ к одному файлу из разных мест. Linux поддерживает *гибкие* (обычно называемые *символьными*) и *жесткие ссылки*.

Когда вы пытаетесь открыть *символьную ссылку*, указывающую на файл, или перейти по ссылке, указывающей на каталог, выполняемая команда перенаправит вас к соответствующему файлу или каталогу. Объект ссылки обладает собственными правами доступа и правами собственности, которые не могут быть определены на основании содержащейся в символьной ссылке информации. Символьная ссылка не обязательно должна располагаться на том же разделе диска, что и сам объект. На самом деле символьная ссылка может существовать даже при отсутствии объекта.

Жесткая ссылка, напротив, может существовать только для файлов (не каталогов) и обычно является одним из способов присвоения имени конкретному физическому файлу. Каждый файл обладает по меньшей мере одной символьной ссылкой, под которой обычно понимается сам файл. Любые дополнительные имена (жесткие ссылки), указывающие на данный отдельно взятый файл, должны находиться на том же разделе, что и сам файл-объект ссылки (по большому счету,

чтобы определить, что файлы являются жесткими ссылками, достаточно выявить, что они обладают одним и тем же номером inode). Изменение прав доступа, прав собственности, отметок даты и времени или содержимого любой жесткой ссылки вызовет соответствующие изменения и в самом файле, однако удаление ссылки не повлечет за собой удаление самого файла — он будет существовать до тех пор, пока не будет удалена последняя жесткая ссылка на него.

Ниже приведено несколько примеров использования команды ln для **создания жестких и символьных ссылок**:

```
$ touch myfile
$ ln myfile myfile-hardlink
$ ln -s myfile myfile-symlink
$ ls -li myfile*
292007   -rw-r--r--   3 francois   francois 0 Mar 25 00:07 myfile
292007   -rw-r--r--   3 francois   francois 0 Mar 25 00:07 myfile-hardlink
292008   lrwxrwxrwx   2 francois   francois 6 Mar 25 00:09 myfile-symlink
```

Стоит отметить, что здесь для отображения результатов после создания жестких и символьных ссылок была использована команда ls -li. Параметр -li отображает номера inode, сопоставляемые с каждым файлом. Так, вы можете видеть, что и myfile, и myfile-hardlink имеют одни и те же номера inode — 292007 (точно обозначающие файл на жестком диске). Символьная ссылка myfile-symlink имеет уже другой номер inode, и, хотя жесткая ссылка фигурирует просто как файл (знак -), символьная ссылка обозначена уже как ссылка (l) с полностью открытыми правами доступа. Вы не сможете определить, разрешен ли вам доступ к файлу, на который указывает символьная ссылка, пока вы не перейдете по ней или не просмотрите сам файл.

Файлы устройств

Когда приложениям необходима связь с устройствами компьютера, они направляют данные в файлы соответствующих устройств. По умолчанию *файлы устройств* хранятся в папке /dev. Сами же устройства обычно делятся на блочные (например, хранилища) и идентификаторы накопителя (например, последовательные порты и конечные устройства).

ПРИМЕЧАНИЕ

Файлы устройств часто называют драйверами. В Linux и UNIX операционная система почти все интерпретирует как файл, откуда и термин «файлы устройств».

Каждый файл устройства связывается с технологическим устройством (при этом указывается тип этого устройства) и номером устройства (указывается номер экземпляра устройства). Например, конечные устройства представлены технологическим устройством под номером 4, в то время как жесткие диски SCSI представлены технологическим блочным устройством под номером 8. Вот **несколько примеров файлов устройств**:

```
$ ls -l /dev/tty0 /dev/sda1 Отображает специальные идентификаторы накопителя
brw-rw---- 1 root disk 8, 1 2007-09-05 08:34 /dev/sda1
crw-rw---- 1 root root 4, 0 2007-09-05 08:34 /dev/tty0
```

Чтобы получить информацию о команде MAKEDEV, предназначенной для отображения списка номеров и имен устройств в Ubuntu Linux, ознакомьтесь с онлайн-страницей справочника, посвященной этой команде. Большинство файлов устройств создается автоматически во время загрузки, поэтому большинство людей никогда вручную не создает файлы устройств. Тем не менее, воспользовавшись командой mknod, вы можете **создать собственный файл устройства**:

```
$ sudo mknod /dev/ttyS4 c 4 68     Добавляет устройство для пятого
                                   серийного порта
$ ls -l /dev/ttyS4                 Отображает список новых файлов устройств
crw-r--r-- 1 root root 4, 68 Sep 6 00:35 /dev/ttyS4
```

Именованные каналы и сокеты

Если вам необходимо, чтобы информация из одного процесса передавалась в другой, достаточно передать выводимые данные одного процесса во вход другого. Однако, чтобы обеспечить эффект присутствия в файловой системе, из которой один процесс может осуществлять связь с другим, предназначены *именованные каналы* и *сокеты*. Именованные каналы обычно используются для осуществления связи между процессами, в то время как сокеты — для обеспечения связи в сети.

Именованные каналы и сокеты часто размещаются приложениями в папке /tmp. Ниже приведено **несколько примеров именованных каналов и сокетов**:

```
$ ls -l /tmp/.TV-chris/tvtimefifo-local /tmp/.X11-unix/X0
prw------- 1 chris chris 0 Sep 26 2007 /tmp/.TV-chris/tvtimefifo-local
srwxrwxrwx 1 root chris 0 Sep 4 01:30 /tmp/.X11-unix/X0
```

В первом случае речь идет об именованном канале, установленном в tvtime TV card player (символ p в начале означает, что это именованный канал). Во втором случае речь идет о сокете, установленном GUI X для связи между процессами.

Чтобы **создать собственный именованный канал**, воспользуйтесь командой mkfifo:

```
$ mkfifo mypipe
$ ls -l mypipe
prw-r--r-- 1 chris chris 0 Sep 26 00:57 mypipe
```

Установление прав доступа к файлам и папкам

Возможности доступа к файлам, выполнения команд и перехода к каталогам могут быть ограничены настройками прав доступа для пользователя, группы пользователей и др. При составлении полного списка файлов и каталогов в Linux (команда ls -l) первые 10 отображаемых символов определяют сам элемент (файл, каталог, блочное устройство и т. д.), с которым или без которого этот элемент может быть прочитан, записан и/или выполнен. На рис. 4.1 показаны значения этих символов.

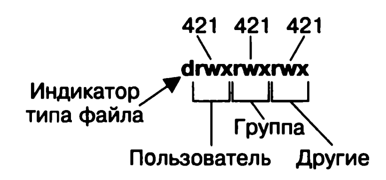


Рис. 4.1. Права доступа, устанавливаемые для файлов и каталогов

Для выполнения примеров, приведенных в данном разделе, создайте каталог /tmp/test и файл /tmp/test/hello.txt, а затем отобразите оба эти элемента:

```
$ mkdir /tmp/test
$ echo "some text" > /tmp/test/hello.txt
$ ls -ld /tmp/test/ /tmp/test/hello.txt
drwxr-xr-x 2 francois sales 4096 Mar 21 13:11 /tmp/test
-rw-r--r-- 2 francois sales 10 Mar 21 13:11 /tmp/test/hello.txt
```

Первый символ полученного списка указывает на то, что /tmp/test является каталогом (d), а hello.txt — файлом (-). Другими типами файлов, доступными для Linux и определяемыми первым символом, являются идентификаторы накопителя (c), блочные устройства (b) или символьные ссылки (l), именованные каналы (p) и сокеты (s).

Следующие девять символов определяют права доступа к файлу и каталогу. Первая группа символов rwx обозначает, что владельцу (francois) предоставлены права на чтение, запись и выполнение файлов в данном каталоге. Аналогичным образом можно сказать, что группа sales обладает более ограниченным доступом (r-x) к каталогу без права записи в него. Все остальные пользователи также обладают правами только на чтение и выполнение (r-x) — дефис обозначает отсутствие прав записи. Что же касается файла hello.txt, то здесь пользователь обладает правами на чтение и запись (rw-), а пользователи группы и все остальные — только на чтение (r--).

При изменении прав доступа к элементам каждое значение может быть представлено в виде восьмеричного числа (чтение — 4, запись — 2, а выполнение — 1) или буквенно (rwx). Вообще, право на чтение предоставляет возможность просматривать содержимое каталога, на запись — изменять (добавлять или модифицировать) его, а на выполнение — переходить (другими словами, получать доступ) к нему.

Если вас не устраивают настройки прав доступа к известным вам файлам или каталогам, то можете поменять их с помощью команды chmod.

зменение прав доступа с помощью команды ımod

Команда chmod позволяет **изменять права доступа к файлам и каталогам**. В табл. 4.1 приведены некоторые примеры использования команды chmod, а также получения доступа к каталогам и изменения файлов.

Таблица 4.1. Изменение прав доступа к файлам и каталогам

Команда chmod (восьмеричным числом или буквами)	Оригинальные права доступа	Новые права доступа	Описание
chmod 0700	any	drwx------	Владелец может читать, записывать файлы в данный каталог, а также переходить в него. Все остальные пользователи (за исключением суперпользователя) не имеют к нему доступа
chmod 0711	any	drwx--x--x	Владелец обладает теми же правами. Все остальные могут открывать каталог, но не могут просматривать или изменять файлы в нем. Это может быть полезно для усиления защиты сервера, когда необходимо запретить просмотр содержимого каталога, но требуется открыть доступ к отдельному файлу
chmod go+r	drwx------	drwxr--r--	Добавление прав чтения каталога может привести к нежелательным результатам, однако если не выполнить этого действия, другие не смогут просматривать файлы, содержащиеся в нем
chmod 0777 chmod a=rwx	any	drwxrwxrwx	Полный доступ
chmod 0000 chmod a-rwx	any	d---------	Доступ полностью закрыт. Данные изменения могут пригодиться, если необходимо защитить каталог от случайных изменений, однако при этом программам резервного копирования, запущенным обычным пользователем, может не удасться осуществить резервное копирование содержимого каталога
chmod 666	any	-rw-rw-rw-	Открывает права на чтение и изменение файла
chmod go-rw	-rw-rw-rw-	-rw-------	Не позволяет никому, кроме самого владельца, просматривать, вносить изменения или удалять файл
chmod 644	any	-rw-r--r--	Только владелец может вносить изменения в файл или удалять его, но все могут его просматривать

Первый ноль в строке режима обычно опускается (то есть допускается использование 777 вместо 0777). Этот структурный ноль имеет специальное значение: это восьмеричная цифра, которая может использоваться в командах (исполнительных) для обозначения того, что данная команда может выполняться как программа-установщик идентификатора пользователя (UID) (4), программа-установщик идентификатора группы (GID) (2) или стать *битом закрепления в памяти* (1).

С программами-установщиками UID и GID команда выполняется с правами доступа, назначенными для пользователей или групп (а не с правами пользователя или группы, запустивших команду).

ВНИМАНИЕ

SUID не должны использоваться в сценариях командного процессора. В разделе справки Linux, посвященном безопасности, содержится следующее предупреждение: «SUID-сценарии командного процессора представляют собой серьезную угрозу безопасности, и именно по этой причине ядро не принимает их. Каким бы безопасным вы не считали сценарий командного процессора, он может эксплуатироваться взломщиками для получения ими прав суперпользователя».

Наличие активного для каталога бита закрепления в памяти ограждает пользователей от угрозы переноса или переименования файлов, хранящихся в каталоге, владельцами которого они не являются (например, /tmp). Однако, если указать верные настройки прав доступа, пользователи смогут изменять содержимое файлов каталога с битом закрепления в памяти, владельцами которых они не являются. Последним символом, используемым для определения прав доступа, является t (вместо x в каталоге бита закрепления в памяти). Ранее команда с активным битом закрепления оставалась в памяти, даже когда не использовалась. Это старая характеристика UNIX, не поддерживаемая более в Linux.

Параметр -R является полезным элементом команды chmod. С его помощью можно **рекурсивно изменять права доступа ко всем файлам и каталогам, начиная с определенной точки файловой системы**:

```
$ sudo chmod -R 700 /tmp/test     Предоставляет полные права доступа
                                  только к содержимому каталога /tmp/test
$ sudo chmod -R 000 /tmp/test     Снимает все права доступа
                                  к содержимому каталога /tmp/test
$ sudo chmod -R a+rwx /tmp/test   Предоставляет полные права доступа
                                  ко всему содержимому каталога /tmp/test
```

Следует отметить, что параметр -R включается в указанный вами каталог. Таким образом, вышеуказанные права доступа, например, заменятся на права доступа к каталогу /tmp/test, а не только к файлам и каталогам, содержащимся в нем.

Команда umask

Права доступа к файлу или каталогу обычно определяются при создании соответствующего элемента. Способ назначения этих прав основывается на текущем значении *umask* пользователя. Используя команду umask, вы можете установить права доступа к файлам и каталогам при их создании.

```
$ umask 0066   Создает каталоги с правами доступа drwx--x--x и файлы
               с правами доступа -rw-------
$ umask 0077   Создает каталоги с правами доступа drwx------ и файлы
               с правами доступа -rw-------
$ umask 0022   Создает каталоги с правами доступа drwxr-xr-x и файлы
               с правами доступа -rw-r--r--
$ umask 0777   Создает каталоги с правами доступа d--------- и файлы
               с правами доступа ----------
```

Изменение прав собственности

Когда вы создаете файл или каталог, на него назначается ваша учетная запись. Это и есть ваша основная группа. Будучи суперпользователем, вы можете, используя команды chown и chgrp, **изменять права собственности (пользователя) и группу, назначенные файлу**:

```
$ chown chris test/          Изменяет владельца на chris
$ chown chris:market test/   Изменяет владельца на chris, а группу на market
$ chgrp market test/         Изменяет группу на market
$ chown -R chris test/       Изменяет владельца всех вложенных файлов
                             и каталогов папки test/ на chris
```

Описанный выше рекурсивный параметр команды chown (-R) полезен, если необходимо изменить права собственности всего дерева каталогов. Как и в случае с командой chmod, рекурсивное использование chown изменяет права доступа для названного каталога и всего его содержимого. Обычно рекурсивное использование команды chown применяется, если кто-то уходит из компании или прекращает пользоваться вашим интернет-сервисом. В этом случае с помощью параметра -R команды chown можно изменить владельца их общего или домашнего каталога на другого пользователя.

Навигация по файловой системе

Основные команды, предназначенные для перемещения по каталогам (cd), проверки текущего каталога (pwd), просмотра содержимого каталога (ls), хорошо известны даже непрофессиональным пользователям командной консоли. Этот раздел посвящен некоторым менее известным параметрам этих команд, а также особенностям навигации по файловой системе. Вот несколько примеров использования команды cd **для навигации по файловой системе**:

```
$ cd                Переходит в ваш основной каталог
$ cd $HOME          Переходит в ваш основной каталог
$ cd ~              Переходит в ваш основной каталог
$ cd ~francois      Переходит в основной каталог пользователя francois
$ cd -              Переходит в предыдущий рабочий каталог
$ cd $OLDPWD        Переходит в предыдущий рабочий каталог
$ cd ~/public_html  Переходит к public_html в вашем основном каталоге
$ cd ..             Переходит в корень рабочего каталога
$ cd /usr/bin       Переходит в каталогу usr/bin из корневого каталога
$ cd usr/bin        Переходит ко вложенной папке usr/bin рабочего каталога
```

Если вы хотите **узнать, какой каталог является рабочим**, воспользуйтесь командой pwd:

```
$ pwd
/home/francois
```

Создание *символьных ссылок* является одним из способов получить доступ к файлу из других частей файловой системы (для получения более подробной

информации о символьных и жестких ссылках обратитесь к подразд. «Символьные и жесткие ссылки»). Однако символьные ссылки могут вызвать некоторую путаницу при отображении корневых каталогов. Следующие команды **создают** в каталоге /tmp **символьную ссылку** на ваш основной каталог и демонстрируют способ определения типа связи с каталогом, на который производится ссылка:

```
$ cd $HOME
$ ln -s /tmp tmp-link
$ ls -l tmp-link
lrwxrwxrwx 1 francois francois 13 Mar 24 12:41 tmp-link -> /tmp
$ cd tmp-link/
$ pwd
/home/francois/tmp-link
$ pwd -P
/tmp
$ pwd -L
/home/francois/tmp-link
$ cd -L ..
$ pwd
/home/francois
$ cd tmp-link
$ cd -P ..
$ pwd
/
```

Использование параметров -P и -L для команд pwd и cd позволяет **работать с каталогами, связанными символьными ссылками, из их основных месторасположений или ссылок на них** соответственно. Например, команда cd -L .. перемещает вас на один уровень вверх относительно вашего основного каталога, тогда как команда cd -P .. перемещает вас на один уровень выше корневого каталога (/). Аналогично параметры -P и -L команды pwd отображают основные месторасположения каталогов и ссылки на них.

Bash может запоминать и хранить список рабочих каталогов. Этот список может быть полезен, если вы захотите вернуться к ранее открываемым каталогам. Для **добавления и удаления каталогов** из этого списка используйте команды pushd и popd:

```
$ pwd
/home/francois
$ pushd /usr/share/man/
/usr/share/man ~
$ pushd /var/log/
/var/log /usr/share/man ~
$ dirs
/var/log /usr/share/man ~
$ dirs -v
 0 /var/log
 1 /usr/share/man
 2 ~
```

```
$ popd
/usr/share/man ~
$ pwd
/usr/share/man
$ popd
~
$ pwd
/home/francois
```

Команды dirs, pushd и popd также могут быть использованы для управления порядком отображения каталогов в стеке. Например, команда pushd -0 перемещает последний каталог из стека на самый верх (делая его рабочим), команда pushd -2 перемещает третий каталог снизу стека на его верх и т. д.

Копирование файлов

Если вы обладаете правами доступа на запись в какой-либо каталог, то копирование файлов и каталогов может осуществляться с помощью нескольких совсем простых команд. Стандартная команда cp **осуществляет копирование файла, сохраняя его имя или присваивая новое**, в новый каталог и создает ему новую отметку времени. Другие параметры команды cp позволяют сохранять отметки даты и времени, осуществлять рекурсивное копирование и запрашивать подтверждение на перезапись:

```
$ cd ; touch index.html
$ mkdir /tmp/html
$ cp -i index.html /tmp/html/
$ cp -il index.html /tmp //html
$ mkdir /tmp/back
$ cp -a /tmp /html /mp/back/
$ cp -R /tmp /html /tmp/back/
```

В приведенных примерах продемонстрированы способы копирования файлов. В первом примере использования команды cp, если файл index.html уже существует в папке /tmp/html, перед перезаписью поверх него нового файла отобразится запрос на подтверждение этого действия. В следующем примере файл index.html является объектом жесткой ссылки, имеющей то же имя и хранящейся в каталоге /tmp/html. В данном случае, поскольку обе жесткие ссылки указывают на один и тот же файл, редактирование файла через любую ссылку повлечет за собой изменение содержимого оригинального файла, где бы он ни находился (ссылка может работать только в том случае, если каталог /tmp/html и ваш основной каталог находятся в одной и той же файловой системе).

Команда cp -a копирует все файлы из каталога /tmp/html, сохраняя все настройки прав собственности и доступа. Если, например, файл /tmp/back будет представлен запоминающим USB-устройством, то с помощью этой команды можно будет записать содержимое вашего интернет-сервера на данное запоминающее устройство. Параметр -R осуществляет рекурсивное копирование структуры каталога

и назначает в качестве владельца каталога текущего пользователя, изменяя текущие отметки даты и времени.

Команда dd также предназначена для **копирования данных**. Она весьма эффективна, поскольку в системах Linux все элементы рассматриваются как файлы, включая периферийные устройства, например:

```
$ dd if=/dev/zero of=/tmp/mynullfile count=1
1+0 records in
1+0 records out
512 bytes (512 B) copied, 0.000308544 s, 1.7 MB/s
```

Файл /dev/zero является специальным файлом, генерирующим нулевые символы. В предыдущем примере команда dd использовала файл /dev/zero в качестве файла входящих данных и выводила данные в файл /tmp/mynullfile. Единицей измерения здесь является количество блоков. По умолчанию размер одного блока составляет 512 байт. Таким образом, в результате выполнения данной команды получился файл размером 512 байт, содержащий только нули. Для просмотра содержимого файла можно воспользоваться командой less или vi, однако лучшим приложением для просмотра файла в данном случае будет команда od:

```
$ od -vt x1 /tmp/mynullfile        Просматривает восьмеричный дамп файла
```

Вот еще один пример использования команды dd:

```
$ dd if=/dev/zero of=/tmp/mynullfile count=10 bs=2
10+0 records in
10+0 records out
20 bytes (20 B) copied, 0.000595714 s, 33.6 kB/s
```

На этот раз размер блока был установлен равным 2 байтам, а скопировано было 10 блоков (20 байт). Следующая командная строка **копирует первый раздел основного жесткого IDE-диска** на второй раздел подчиненного жесткого IDE-диска (прежде чем приступить к выполнению подобной операции, выполните резервное копирование данных):

```
$ sudo dd if=/dev/hda1 of=/dev/hdb2
```

ВНИМАНИЕ

Будьте предельно осторожны при использовании этой команды. Вообще, у вас не должно возникнуть необходимости перезаписывать части жестких дисков.

Следующий пример демонстрирует **резервное копирование ведущего жесткого диска IDE со сжатием первого раздела**. Обычно перед подобным копированием раздел демонтируется.

```
$ sudo umount /dev/hda1
$ sudo dd if=/dev/hda1 | gzip > bootpart.gz
```

Следующая команда копирует файл-образ ISO с компакт-диска или DVD на USB-носитель (предполагается, что носитель отображен как /dev/sdb1):

```
$ sudo dd if=whatever.iso of=/dev/sdb1
```

Стоит отметить, что данная команда создает бинарную копию байтов файла, что, возможно, не соответствует вашим целям.

В следующем примере главная загрузочная запись копируется из основного жесткого диска IDE в файл mymbrfile:

```
$ dd if=/dev/hda of=mymbrfile bs=512 count=1
```

Если вам необходимо создать копию образа ISO, записанного на компакт-диск или DVD, вставьте диск в CD/DVD-привод и выполните следующую команду (необходимо, чтобы файл /dev/cdrom соответствовал вашему CD-приводу):

```
$ dd if=/dev/cdrom of=whatever.iso
```

ПРИМЕЧАНИЕ

Помимо файлов устройств /dev/cdrom, Ubuntu создает файлы устройств /dev/cdrw и /dev/dvd.

Изменение атрибутов файла

Все файлы и каталоги в Linux обладают определенными правами доступа на чтение, запись и выполнение, основанными на имени пользователя, группы и др. Однако существуют также другие, присущие только некоторым типам файловых систем атрибуты, которые могут закрепляться за файлами и каталогами.

В файловых системах ext2 и ext3 файлы обладают специальными атрибутами, которые могут использоваться по выбору. Команда lsattr позволяет **просмотреть эти атрибуты**. Большинство атрибутов являются скрытыми и не назначаются по умолчанию. Ниже приведен пример использования команды lsattr для просмотра некоторых атрибутов файлов:

```
$ lsattr /etc/host*
------------- /etc/host.conf
------------- /etc/hosts
------------- /etc/host.allow
------------- /etc/host.deny
$ lsattr -aR /tmp/ | less   Рекурсивно отображает все атрибуты каталога /tmp
```

Дефисы соответствуют 13 атрибутам ext2/ext3, которые могут быть установлены. Ни один из них не является отображаемым по умолчанию: a (только добавление), c (сжатый), d (без дампа), I (постоянный), j (регистрация данных), s (безопасное удаление), t (запрет слияния в конце файла), u (неудаляемый), A (без обновления atime), D (синхронные обновления каталогов), S (синхронные обновления) и T (верхушка дерева каталогов). С помощью команды chattr вы можете изменять эти атрибуты:

```
$ sudo chattr +i whatever.iso
$ sudo chattr +A -R /home/francois/images/*
$ sudo chattr +d ubuntu-7.04-desktop.i386.iso
$ lsattr whatever.iso /home/francois/images/* ubuntu-7.04-desktop.i386.iso
----i-------- whatever.iso
-------A----- /home/francois/images/einstein.jpg
-------A----- /home/francois/images/goth.jpg
------d------ ubuntu-7.04-desktop.i386.iso
```

Как видно из предыдущего примера, при использовании параметра +i файл whatever iso становится постоянным, то есть он не может быть удален, переименован или изменен, а также для него не может быть создана ссылка. Это предотвращает любые случайные изменения файла (даже суперпользователь не сможет внести изменения в файл, пока не будет снят атрибут i.) Используйте этот атрибут для обеспечения безопасности системных файлов.

Параметр -R, приведенный в примере, рекурсивно устанавливает параметр +A, что запрещает всем файлам, хранящимся в каталоге images и его подкаталогах, изменять время доступа (atime). Установка атрибута A может помочь сэкономить I/O дисков ноутбуков и флэш-накопителей. Если вы для резервного копирования своих файловых систем ext2/ext3 используете команду dump, то к файлам, обладающим атрибутом d, копирование применено не будет. В данном случае мы предпочитаем не выполнять копирования больших образов ISO.

Для **удаления атрибута** воспользуйтесь командой chatter совместно со знаком -:

```
$ sudo chattr -i whatever.iso
```

ИМЕЧАНИЕ

Взломщики, которым успешно удалось проникнуть в систему, часто заменяют некоторые системные бинарные файлы (например, ls или ps) поврежденными версиями и делают их неотключаемыми. Поэтому полезно иногда проверять атрибуты, назначенные исполняемым файлам (например, в каталогах /bin, /usr/bin, /sbin и /usr/sbin).

оиск файлов

Ubuntu с помощью приложений из пакета mlocate создает базу данных всех файлов файловой системы (с несколькими исключениями, заданными в файле /etc/updatedb.conf). Команда locate позволяет производить поиск по этой базе данных (в Ubuntu команда locate представляет собой символьную ссылку на команду slocate). Результаты выводятся после завершения поиска по базе данных (а не по самой файловой системе). До внедрения же команды locate большинство пользователей Linux для поиска файлов в файловой системе использовали команду find. Ниже описаны как команда locate, так и команда find.

оиск файлов с помощью команды locate

Поскольку в базе данных присутствуют имена всех элементов файловой системы, а не только команды, вы можете использовать locate для **поиска команд, устройств, страниц справочника man, файлов данных и любого другого элемента файловой системы, распознаваемого по имени**:

```
$ locate e1000
/lib/modules/2.6.20-16-generic/kernel/drivers/net/e1000
/lib/modules/2.6.20-16-generic/kernel/drivers/net/e1000/e1000.ko
/lib/modules/2.6.20-15-generic/kernel/drivers/net/e1000
/lib/modules/2.6.20-15-generic/kernel/drivers/net/e1000/e1000.ko
/usr/src/linux-headers-2.6.20-16-generic/include/config/e1000
/usr/src/linux-headers-2.6.20-16-generic/include/config/e1000/napi.h
/usr/src/linux-headers-2.6.20-16-generic/include/config/e1000.h
```

```
/usr/src/linux-headers-2.6.20-15-generic/include/config/e1000
/usr/src/linux-headers-2.6.20-15-generic/include/config/e1000/napi.h
/usr/src/linux-headers-2.6.20-15-generic/include/config/e1000.h
/usr/src/linux-headers-2.6.20-15/include/config/e1000.h
/usr/src/linux-headers-2.6.20-15/drivers/net/e1000
/usr/src/linux-headers-2.6.20-15/drivers/net/e1000/Makefile
/usr/src/linux-headers-2.6.20-16/include/config/e1000.h
/usr/src/linux-headers-2.6.20-16/drivers/net/e1000
/usr/src/linux-headers-2.6.20-16/drivers/net/e1000/Makefile
```

В приведенном выше примере было найдено две версии модулей ядра e1000.ko. Стоит заметить, что команда locate, если не использовать параметр -i, является чувствительной к регистру:

```
$ locate -i itco_wdt
/lib/modules/2.6.20-16-generic/kernel/drivers/char/watchdog/iTCO_wdt.ko
/lib/modules/2.6.20-15-generic/kernel/drivers/char/watchdog/iTCO_wdt.ko
```

Пакет slocate (в некоторых дистрибутивах Linux называемый mlocate) включает в себя cron job, который для обновления базы данных файлов locate ежедневно выполняет команду updatedb.

Чтобы **немедленно обновить базу данных locate**, вручную выполните команду updatedb:

```
$ sudo updatedb
```

Определение местонахождения файлов с помощью команды find

До внедрения команды locate поиск файлов производился с помощью команды find. Хотя команда locate осуществляет поиск быстрее, у find есть множество других эффективных параметров, полезных для выполнения поиска файлов, основывающегося не на имени, а на других атрибутах.

ПРИМЕЧАНИЕ

Сканирование целой файловой системы может занять достаточно много времени, поэтому, прежде чем приступить к нему, попробуйте просканировать подсистему файловой системы или исключить некоторые каталоги или удаленно смонтированные файловые системы.

В следующем примере выполняется рекурсивный поиск файла с именем e100 в корневой файловой системе (/):

```
$ find / -name "e100*" -print
find: /usr/lib/audit: Permission denied
find: /usr/libexec/utempter: Permission denied
/sys/module/e100
/sys/bus/pci/drivers/e100
...
```

Результатом выполнения команды find от имени обычного пользователя может стать длинный список сообщений Permission denied (Отказано в доступе), поскольку команда find пытается просмотреть каталоги, к которым вы не имеете прав

доступа. Для избежания этого вы можете **исключить из поиска недоступные каталоги**:

```
$ find / -name e100 -print 2>&1 | grep -v "Permission denied"
```

Или отправить все сообщения об ошибках в битоприемник («корзину») /dev/null:

```
$ find / -name e100 -print 2> /dev/null
```

Поскольку команда find является чувствительной к регистру и имя необходимо вводить точно (поиск имен e100 и e100.ko даст различные результаты), вы можете **для более точного поиска использовать регулярные выражения**:

```
$ find / -name 'e100*' -print
/lib/modules/2.6.20-16-generic/kernel/drivers/net/e1000
/lib/modules/2.6.20-16-generic/kernel/drivers/net/e1000/e1000.ko
/lib/modules/2.6.20-16-generic/kernel/drivers/net/e100.ko
/lib/modules/2.6.20-15-generic/kernel/drivers/net/e1000
/lib/modules/2.6.20-15-generic/kernel/drivers/net/e1000/e1000.ko
/lib/modules/2.6.20-15-generic/kernel/drivers/net/e100.ko
/usr/src/linux-headers-2.6.20-16-generic/include/config/e100.h
/usr/src/linux-headers-2.6.20-16-generic/include/config/e1000
/usr/src/linux-headers-2.6.20-16-generic/include/config/e1000.h
/usr/src/linux-headers-2.6.20-15-generic/include/config/e100.h
/usr/src/linux-headers-2.6.20-15-generic/include/config/e1000
/usr/src/linux-headers-2.6.20-15-generic/include/config/e1000.h
/usr/src/linux-headers-2.6.20-15/include/config/e100.h
/usr/src/linux-headers-2.6.20-15/include/config/e1000.h
/usr/src/linux-headers-2.6.20-15/drivers/net/e1000
/usr/src/linux-headers-2.6.20-16/include/config/e100.h
/usr/src/linux-headers-2.6.20-16/include/config/e1000.h
/usr/src/linux-headers-2.6.20-16/drivers/net/e1000
```

Кроме того, вы можете **осуществлять поиск файлов по временным меткам**. Следующая команда осуществляет поиск файлов в каталоге /usr/bin/, к которым обращались в последние две минуты:

```
$ find /usr/bin/ -amin -2 -print
/usr/bin/
/usr/bin/find
```

Следующая же команда ищет в каталоге /home/chris файлы, к которым не обращались более 60 дней:

```
$ find /home/chris/ -atime +60
```

Для **поиска каталогов** предназначен параметр -type d. Следующая команда ищет все подкаталоги /etc и перенаправляет stderr в корзину (/dev/null):

```
$ find /etc -type d -print 2> /dev/null
```

Данная команда находит в каталоге /sbin файлы с правами доступа, соответствующими 750:

```
$ find /sbin/ -perm 750 -print
```

Параметр exec команды find является весьма эффективным, поскольку он позволяет воздействовать на найденные с помощью команды find файлы. Следующая команда ищет в каталоге /var все файлы, владельцем которых является francois (должен быть зарегистрированным пользователем), и выполняет для каждого из них команду ls -l:

```
$ find /var -user francois -exec ls -l {} \;
```

Альтернативным параметру exec команды find является параметр xargs:

```
$ find /var -user francois -print | xargs ls -l
```

В действиях данных команд есть существенные различия, поэтому и результаты их выполнения очень разные. Команда find -exec применяет для каждого полученного результата поиска ls. Команда же xargs работает более эффективно, поскольку в качестве вводимых данных для одной команды ls применяет множество результатов поиска.

Чтобы **инвертировать критерий поиска**, поставьте перед этим критерием восклицательный знак (!). В следующем примере выполняется поиск всех файлов, которые не являются собственностью группы root и являются обычными файлами, а затем для каждого из них выполняется команда ls -l:

```
$ find / ! -group root -type f -print 2> /dev/null | xargs ls -l
```

В следующем примере осуществляется поиск файлов в каталоге /sbin, которые являются обычными файлами с закрытыми правами доступа на запись, а затем для каждого из них применяется команда ls -l:

```
$ find /sbin/ -type f ! -perm /o+w -print | xargs ls -l
-rwxr-xr-x 1 root root   3056  2007-03-07 15:44 /sbin/acpi_available
-rwxr-xr-x 1 root root   43204 2007-02-18 20:18 /sbin/alsactl
```

Поиск файлов определенного размера является прекрасным способом определить, что занимает место на ваших жестких дисках. Следующая команда ищет все файлы, размер которых превышает 10 Мбайт (+10M), составляет список этих файлов в порядке от самых крупных до наиболее мелких (ls -lS) и направляет этот список в текстовый файл (/tmp/bigfiles.txt):

```
$ find / -xdev -size +10M -print | xargs ls -lS > /tmp/bigfiles.txt
```

В данном примере параметр -xdev запрещает поиск в любых смонтированных файловых системах, кроме файловой системы суперпользователя. Это способ позволяет запретить команде find производить поиск в каталоге /proc и любой удаленно или локально смонтированной файловой системе.

Другие команды для поиска файлов

Среди других команд, предназначенных для поиска файлов, можно назвать команды whereis и which:

```
$ whereis man
man: /usr/bin/man /usr/X11R6/bin/man /usr/bin/X11/man /usr/local/man
/usr/share/man /usr/share/man/man1/man.1.gz /usr/share/man/man7/man.7.gz
```

```
$ which ls
/bin/ls
```

Команда whereis позволяет осуществлять поиск не только команд, но и страниц справочника man, а также конфигурационных файлов, связанных с соответствующей командой. Как видно из приведенного выше примера, запрос команде whereis на поиск слова man возвращает исполняемый файл man, его конфигурационный файл, а также месторасположение MAN-страниц для команды man. Второй пример (команда which) отображает адрес исполнительного файла ls (/bin/ls). Команда which полезна, если необходимо определить расположение исполняемого файла в переменной PATH:

```
$ dpkg-query -S `which ps`
procps: /bin/ps
```

Получение более подробной информации о файлах

Теперь, когда вам известно, каким образом можно искать файлы, вы можете получать более подробную информацию о них. Использование менее общих параметров команды ls позволяет отображать информацию о файле, которую вы не могли бы увидеть без их использования. Такие команды, как file, помогают определять тип файла, а используя md5sum и sha1sum, можно проверять подлинность файла.

Отображение списка файлов

Даже если вы хорошо знакомы с командой ls, вам могут быть незнакомы многие полезные параметры этой команды, которые позволяют получать большое количество информации, касающейся файлов вашей системы. Ниже приведено несколько примеров использования команды ls для **отображения длинных списков** (-l) файлов и каталогов:

```
$ ls -l  Отображает список файлов и каталогов в текущем каталоге
$ ls -la Отображает файлы и каталоги, начинающиеся с точки (.)
$ ls -lt Сортирует файлы по времени последнего изменения
$ ls -lu Сортирует файлы по времени последнего открытия
$ ls -lS Сортирует файлы по размеру
$ ls -li Отображает список номеров inode, связанных с каждым файлом
$ ls -ln Отображает вместо имен числовые идентификаторы пользователя/группы
$ ls -lh Отображает размеры файлов в удобной для чтения форме (Кбайт, Мбайт и т. д.)
$ ls -lR Рекурсивно отображает файлы из рабочего каталога и его подкаталогов
```

Существуют также способы **по-разному отображать различные типы файлов** в процессе их просмотра:

```
$ ls -F                    Добавляет символ для определения типа файла
myfile-symlink@ config/    memo.txt pipefile| script.sh* xpid.socket=
$ ls --color=always        Выделяет типы файлов различными цветами
$ ls -C                    Отображает файлы в столбцах
```

В первом примере на выходе команды отображаются различные типы файлов. Запись myfile-symlink@ указывает на то, что это символьная ссылка на каталог, config/ является обычным каталогом, memo.txt — обычным файлом (без дополнительных символов), pipefile| — именованным каналом (созданным с помощью mkfifo), script.sh* — исполняемым файлом, а xpid.socket= — сокетом. В следующих двух примерах типы файлов выделяются цветами и отображаются в столбцах соответственно.

Проверка файлов

Вместе с программными пакетами и образами компакт-дисков или DVD, доступными в Интернете, часто поставляются файлы SHA1SUM или MD5SUM. Эти файлы содержат контрольные суммы, которые могут быть использованы, чтобы убедиться, что загружаемый файл является подлинным и издан репозиторием.

Ниже приведены примеры использования команд md5sum и sha1sum **для генерации контрольных сумм файлов**:

```
$ md5sum whatever.iso
d41d8cd98f00b204e9800998ecf8427e whatever.iso
$ sha1sum whatever.iso
da39a3ee5e6b4b0d3255bfef95601890afd80709 whatever.iso
```

Выбор команды зависит от того, какую информацию предоставляет поставщик проверяемого вами файла — md5sum или sha1sum. Например, файл md5sum.txt для дистрибутива Ubuntu Feisty содержит следующую информацию:

```
90537599d934967f4de97ee0e7e66e6c ./dists/feisty/main/binary-i386/Release
c53152b488a9ed521c96fdfb12a1bbba ./dists/feisty/main/binary-i386/Packages
ba9a035c270ba6df978097ee68b8d7c6 ./dists/feisty/main/binary-i386/Packages.gz
...
```

Этот файл отображает все контрольные суммы MD5 для всех файлов на Ubuntu 7.04 Live CD.

С помощью параметра -c команды md5sum вы можете **проверить сразу всю информацию о файлах**, которая представлена в файле md5sum.txt:

```
$ md5sum -c md5sum.txt
./dists/feisty/main/binary-i386/Release: OK
./dists/feisty/main/binary-i386/Packages: OK
./dists/feisty/main/binary-i386/Packages.gz: OK...
```

Чтобы **проверить только один файл из списка**, можно выполнить следующую команду:

```
$ cat md5sum.txt | grep Release.gpg |md5sum -c
./dists/feisty/Release.gpg: OK
```

Если для проверки на соответствие у вас вместо файла md5sum.txt есть только файл SHA1SUM, вы можете аналогичным образом использовать команду sha1sum. Используя совместно описанную ранее в данной главе команду find и команду md5sum, вы можете проверять любую часть файловой системы. Например, можно

создать контрольную сумму MD5 для всех файлов каталога /etc таким образом, чтобы позже их можно было проверить на наличие изменений:

```
$ sudo find /etc -type f -exec md5sum {} \; > /tmp/md5.list 2> /dev/null
```

В результате выполнения предыдущей команды создается файл /tmp/md5.list, содержащий 128-битную контрольную сумму для каждого файла каталога /etc. Теперь, чтобы определить, были ли изменены какие-либо из этих файлов, вы можете в любой момент выполнить следующую команду:

```
$ cd /etc
$ md5sum -c /tmp/md5.list | grep -v 'OK'
./hosts.allow: FAILED
md5sum: WARNING: 1 of 1668 computed checksums did NOT match
```

Как вы можете заметить, в данном случае был изменен лишь один файл (hosts.allow), поэтому теперь необходимо будет проверить измененный файл и определить, были ли эти изменения намеренными.

Резюме

В Linux существуют десятки команд, предназначенных для получения информации о файлах и работы с ними. Такие команды, как chmod, могут изменять права доступа к файлам, тогда как команды наподобие lsattr и chattr могут использоваться для просмотра и изменения атрибутов, связанных с такими файловыми системами, как ext2 и ext3.

Для навигации по файловой системе наиболее часто используется команда cd, однако для повторяющегося передвижения по одним и тем же каталогам вы можете использовать команды pushd и popd, позволяющие работать с хранилищем каталогов.

Копирование файлов чаше всего осуществляется с помощью команды cp, однако для копирования файлов (таких как образы дисков) с устройства (например, CD-ROM) может быть использована и команда dd. Для создания каталогов вы можете использовать команду mkdir.

Чтобы не создавать множество копий одного и того же файла, разбросанных по всей системе, вы можете воспользоваться символьными и жесткими ссылками, позволяющими закрепить за одним файлом или каталогом несколько имен. Символьные ссылки могут присутствовать где угодно в системе, в то время как жесткие ссылки должны находиться на одном разделе с файлом-оригиналом.

Для поиска файлов в Linux используются команды locate и find, а для проверки подлинности файлов, загружаемых из Интернета, — команды md5sum и sha1sum.

5 Обработка текстовой информации

Использование первых UNIX-систем (на которых была основана Linux), управляемых только с помощью командной консоли, требовало тесной работы с командами и простыми текстовыми файлами. Документы, исходный текст, файлы конфигурации, электронная почта и почти все создаваемые или настраиваемые вами элементы системы представлены в виде текстовых файлов. Для работы с подобными файлами разработчики того времени создали множество приложений, позволяющих редактировать текст.

Несмотря на наличие графических приложений для работы с текстом, наиболее опытные пользователи Linux считают приложения для командной строки более эффективными и удобными. Текстовые редакторы vi (vim), Emacs, JOE, nano и Pico доступны на большинстве дистрибутивов Linux. Такие команды, как `grep`, `sed` и `awk`, могут быть использованы для поиска и, возможно, изменения фрагментов информации, содержащихся в текстовых файлах.

В этой главе рассказывается о способах использования многих популярных команд, предназначенных для работы с текстовыми файлами в Ubuntu. Кроме того, здесь рассматриваются менее известные способы применения команд по редактированию текста, которые могут показаться вам интересными.

Поиск в тексте с помощью регулярных выражений

Многие приложения, предназначенные для работы с текстом, позволяют использовать *регулярные выражения* (*regular expressions*), иногда называемые *regex*, для поиска необходимого текста на основе определенной схемы. Эти выражения могут использоваться для поиска текста внутри текстового редактора или же, в совокупности с командами поиска, для сканирования большого количества файлов на наличие нужных строк.

Схема поиска regex может основываться на конкретной текстовой строке (или всего одном слове, например `Linux`) или месторасположении (например, конец строки или начало слова). Поиск может быть узким (найти именно слово `hello`)

или более широким (найти любое слово, начинающееся на h и заканчивающееся на o).

Приложение 3 включает в себя справочную информацию о метасимволах командного процессора, которые могут быть использованы в сочетании с регулярными выражениями для осуществления точного поиска. В данном разделе приведены примеры использования регулярных выражений совместно с некоторыми приложениями, которые будут рассмотрены в этой главе.

В табл. 5.1 приведены примеры использования некоторых регулярных выражений для поиска нужных строк в тексте. Многие из представленных здесь примеров используются в примерах на протяжении всей главы.

Помните, что не каждая команда, основывающаяся на regex, одинаково использует его элементы.

Таблица 5.1. Поиск с помощью регулярных выражений

Выражение	Результат поиска
a*	a, ab, abc и aecjejich
^a	Любое «а», находящееся в начале строки
*a$	Любое «а», находящееся в конце строки
a.c	Трехсимвольные строки, начинающиеся с «а» и заканчивающиеся на «с»
[bcf]at	bat, cat или fat
[a-d]at	aat, bat, cat, dat, но не Aat, Bat и т. д.
[A-D]at	Aat, Bat, Cat и Dat, но не aat, bat и т. д.
1[3-5]7	137, 147 и157
\tHello	Символ табуляции, предшествующий слову hello
\.[tT][xX][Tt]	.txt, .TXT, .TxT или другие сочетания на основе изменения регистра

Редактирование текстовых файлов

В мире Linux/UNIX используется множество текстовых редакторов. Наиболее широко используемым из них является редактор vi, который можно найти практически на любой современной операционной системе UNIX. Именно поэтому умение редактировать даже незначительный текстовый файл в vi является обязательным требованием для администратора Linux. Если однажды вам придется восстанавливать онлайн-подключение в незнакомой среде, vi может оказаться приложением, которое всегда будет под рукой.

Убедитесь, что в вашей версии Ubuntu установлен расширенный пакет vim. Редактор vim представляет собой наиболее современный, с большим количеством функций и дружественный пользователю редактор vi. Для получения более подробной информации по использованию vi ознакомьтесь с Приложением 1.

ПРИМЕЧАНИЕ

Ubuntu устанавливает vim по умолчанию.

Традиционно, другим популярным текстовым редактором для UNIX является Emacs и его вариант, более ориентированный на графическую оболочку, — XEmacs. Emacs является мощным многофункциональным приложением, которое также может служить для чтения почты или новостей, а также выполнять другие функции. Emacs также известен своими очень сложными клавиатурными сокращениями, для свободной работы с которыми необходимо иметь три руки.

В середине 1990-х годов Emacs превзошел vi в отношении характеристик. Сейчас же, когда широко распространен vim, оба редактора способны предоставлять любые необходимые функции. Если вы еще недостаточно хорошо знакомы с vi и Emacs, мы рекомендуем вам начать с изучения vi.

В Linux существует много других команд и приложений GUI для редактирования текста. Среди консольных редакторов, которые могут вам показаться проще в обращении, чем vi или Emacs, можно выделить JED, JOE и nano. Запустите любой из этих редакторов, указав в командной строке его название, при желании дополнив его в конце именем файла, который хотите отредактировать. В следующих подразделах дано короткое описание использования каждого из перечисленных выше редакторов.

Использование редактора JOE

Если вы ранее пользовались классическими текстовыми процессорами, например WordStar, работающими с текстовыми файлами, вам будет комфортно работать и в редакторе JOE. Для его использования установите программный пакет JOE. Для использования средства проверки орфографии в JOE убедитесь, что установлен программный пакет aspell (Ubuntu устанавливает его по умолчанию). Для установки JOE выполните следующую команду:

```
$ sudo apt-get install joe
```

Для перемещения по файлу в редакторе JOE предназначены знаки управления и клавиши управления курсором. Чтобы **открыть текстовый файл для редактирования**, просто введите команду `joe` и имя необходимого файла или воспользуйтесь одной из следующих команд:

```
$ joe memo.txt                      Открывает файл memo.txt для редактирования
$ joe -wordwrap memo.txt            Включает автоматический перенос в процессе
                                    редактирования
$ joe -lmargin 5 -tab 5 memo.txt    Устанавливает левую границу
                                    в положение 5, а значение табуляции на 5
$ joe +25 memo.txt                  Начинает редактирование на строке 25
```

Чтобы **добавить текст**, просто начните печатать. Для многих функций редактор поддерживает **клавиатурные сокращения**. Используйте клавиши со стрелками для передвижения курсора влево, вправо, вверх или вниз. Используйте клавишу Delete для удаления текста под курсором или Backspace для удаления текста слева от курсора. Нажмите Enter для переноса строки. Для вызова справки используйте сочетание клавиш Ctrl+K+H. В табл. 5.2 приведены наиболее общие примеры использования сочетаний клавиш для редактирования текста в JOE.

Таблица 5.2. Сочетания клавиш, предназначенные для редактирования в JOE

Сочетание клавиш	Результат использования
Перемещение курсора	
Ctrl+B	Влево
Ctrl+P	Вверх
Ctrl+F	Вправо
Ctrl+N	Вниз
Ctrl+Z	Предыдущее слово
Ctrl+X	Следующее слово
Поиск	
Ctrl+K+F	Найти текст
Ctrl+L	Найти далее
Абзац	
Ctrl+K+B	Переместиться в начало абзаца
Ctrl+K+K	Переместиться в конец абзаца
Ctrl+K+M	Переместить блок
Ctrl+K+C	Скопировать блок
Ctrl+K+W	Записать блок в файл
Ctrl+K+Y	Удалить блок
Ctrl+K+/	Установить фильтр
Разное	
Ctrl+K+A	Центральная строка
Ctrl+T	Параметры
Ctrl+R	Обновить информацию
Файл	
Ctrl+K+E	Открыть новый файл для редактирования
Ctrl+K+R	Вставить файл на место расположения курсора
Ctrl+K+D	Сохранить
Переход	
Ctrl+U	Предыдущий экран
Ctrl+V	Следующий экран
Ctrl+A	Начало строки
Ctrl+E	Конец строки
Ctrl+K+U	Начало файла
Ctrl+K+V	Конец файла
Ctrl+K+L	Перейти к строке номер
Удаление	
Ctrl+D	Удалить символ
Ctrl+Y	Удалить строку
Ctrl+W	Удалить слово справа

Продолжение ⇨

Таблица 5.2 (продолжение)

Сочетание клавиш	Результат использования
Ctrl+O	Удалить слово слева
Ctrl+J	Удалить строку справа
Ctrl+-	Отменить действие
Ctrl+6	Вернуть действие
Выход	
Ctrl+K+X	Сохранить и выйти
Ctrl+C	Отменить
Ctrl+K+Z	Выйти в консоль
Проверка правописания	
Ctrl+[+N	Слово
Ctrl+[+L	Файл

Работа с редакторами Pico и nano

Pico — это популярный небольшой текстовый редактор, распространяемый как часть почтового клиента Pine. Хотя Pico является бесплатной программой, его код не открыт. Именно поэтому многие дистрибутивы Linux, включая Ubuntu, не предоставляют по умолчанию Pico. Вместо него они предлагают клон Pico с открытым исходным кодом, называемый nano (nano — еще один редактор; от англ. another). В этом подразделе описывается именно этот редактор.

ПРИМЕЧАНИЕ

Команда pico в Ubuntu ссылается на редактор nano.

Редактор nano (представленный командой nano) является компактным текстовым редактором, который запускается из командной консоли и рассчитан на работу с экраном (поскольку он ориентирован на работу с регистром). Nano популярен среди тех, кто раньше использовал почтовый клиент Pine, так как nano использует те же инструменты редактирования, что и Pico. В некоторых редких случаях, когда в системе Linux недоступен редактор vi (если, например, вы устанавливаете минимальный набор приложений Gentoo Linux), nano все равно будет доступен. Ubuntu устанавливает nano по умолчанию. Для осуществления проверки правописания в редакторе nano используется команда `spell`, а не `aspell`.

Как и в случае с редактором JOE, редактор nano не требует выполнения специальных команд для начала редактирования — открыв документ, вы можете сразу начинать печатать. Чтобы **открыть текстовый файл для редактирования**, просто введите команду nano и имя файла или воспользуйтесь одним из следующих параметров:

```
$ nano memo.txt        Открывает файл memo.txt для редактирования
$ nano -B memo.txt     При сохранении осуществляет резервное копирование
                       предыдущего варианта в файл ~.имя файла
$ nano -m memo.txt     Включает управление курсором с помощью мыши
                       (если это поддерживается)
$ nano +83 memo.txt    Начинает редактирование со строки 83
```

Параметр -m командной строки активизирует поддержку мыши, которая может быть использована для выбора позиции курсора в тексте. Следует помнить, что, если дважды щелкнуть кнопкой мыши на тексте, будет выделен целый его блок, что не всегда удобно.

Чтобы перемещать курсор влево, вправо, вверх или вниз, используйте клавиши со стрелками. Используйте клавишу Delete для удаления выделенного текста под курсором или Backspace для удаления текста слева от курсора. Для переноса строки нажмите клавишу Enter. Для вызова справки используйте сочетание клавиш Ctrl+G. В табл. 5.3 приведены сочетания клавиш, используемые nano (их описание также содержится в справке nano).

Таблица 5.3. Сочетания клавиш, используемые для редактирования в nano

Сочетание клавиш	Функциональная клавиша	Описание
Ctrl+G	F1	Показывает текст помощи (для выхода нажмите Ctrl+x)
Ctrl+X	F2	Выходит из nano (или закрывает текущий буфер файла)
Ctrl+O	F3	Сохраняет текущий файл
Ctrl+J	F4	Выравнивает текст в текущем абзаце
Ctrl+R	F5	Вставляет текст в текущий файл
Ctrl+W	F6	Запускает поиск в тексте
Ctrl+Y	F7	Переходит к предыдущему экрану
Ctrl+V	F8	Переходит к следующему экрану
Ctrl+K	F9	Вырезает (и помещает в буфер обмена) текущий файл или выделенный текст
Ctrl+U	F10	Вставляет в файл ранее вырезанную строку
Ctrl+C	F11	Отображает текущее положение курсора
Ctrl+T	F12	Начинает проверку правописания
Ctrl+-	–	Переходит к выделенной строке и номерам столбцов
Ctrl+\	–	Ищет и заменяет текст
Ctrl+6	–	Выделяет текст начиная от места расположения курсора (для снятия выделения нажмите Ctrl+6)
Ctrl+F	–	Перемещает курсор на один символ вперед
Ctrl+B	–	Перемещает курсор на один символ назад
Ctrl+Пробел	–	Перемещает курсор на одно слово вперед
Alt+Пробел	–	Перемещает курсор на одно слово назад
Ctrl+P	–	Перемещает курсор на предыдущую строку
Ctrl+N	–	Перемещает курсор на следующую строку
Ctrl+A	–	Перемещает курсор в начало текущей строки
Ctrl+E	–	Перемещает курсор в конец текущей строки
Alt+(	–	Перемещает курсор в начало текущего абзаца
Alt+)	–	Перемещает курсор к концу текущего абзаца
Alt+\	–	Перемещает курсор к первой строке файла
Alt+/	–	Перемещает курсор к последней строке файла

Продолжение ⇨

Таблица 5.3 (продолжение)

Сочетание клавиш	Функциональная клавиша	Описание
Alt+]	–	Перемещает курсор ко второй скобке
Alt+=	–	Прокручивает текст вниз на одну строку
Alt+-	–	Перемещает строку вверх

Графические текстовые редакторы

Сам факт редактирования текста не означает, что вы используете именно консольный редактор. Основные преимущества использования графических текстовых редакторов состоят в том, что вы можете использовать мышь для выбора меню, выделять, вырезать, копировать текст и пользоваться специальными расширениями.

Если на вашей операционной системе Linux установлена графическая оболочка GNOME, то наверняка у вас установлен и текстовый редактор для GNOME (gedit). Инструменты gedit позволяют осуществлять проверку правописания, отображать статистику документа, изменять шрифты и цвета и распечатывать документы. В оболочке Рабочего стола KDE также присутствует собственный текстовый редактор KDE (kedit в программном пакете kdeutils). Он предоставляет тот же набор функций, что и текстовый редактор GNOME, а также некоторые дополнительные, например возможность отправлять документы с помощью kmail или другого конфигурируемого пользователем компонента KDE.

Сам по себе редактор vim входит в версию X GUI. Он запускается с помощью команды `gvim`, являющейся частью программного пакета vim-X11. Если вы хотите сделать vim для GUI более удобным, то загрузите конфигурацию Cream с сайта http://cream.sourceforge.net/.

ПРИМЕЧАНИЕ

Для использования gvim вам придется установить дополнительный пакет — vim-gnome.

Среди других графических текстовых редакторов, которые вы можете установить, можно назвать nedit (предоставляет инструменты для использования макросов и выполнения консольных команд и ориентирован на разработчиков программного обеспечения) и leafpad (предоставляет те же инструменты, что и Windows Notepad). Текстовый редактор Scribes (scribes) предоставляет некоторые специальные инструменты для автоматического исправления, замены, сдвига и дополнения слов.

Отображение, упорядочивание и редактирование текста

Помимо обычного редактирования отдельных текстовых файлов, для отображения, поиска и управления содержимым одного или более текстовых файлов одновременно вы можете использовать большое количество команд, доступных в Linux.

гображение текстовых файлов

Основным способом просмотра содержимого текстового файла является выполнение команды cat. Команда cat конкатенирует (или выводит данные в виде строки символов) содержимое текстового файла на экран (по умолчанию). Чтобы **различными способами выводить содержимое файла**, вы можете использовать различные метасимволы командного процессора:

```
$ cat myfile.txt                        Отображает содержимое файла на экране
$ cat myfile.txt > copy.txt             Отправляет содержимое файла в другой файл
$ cat myfile.txt >> myotherfile.txt     Прикрепляет содержимое файла к другому файлу
$ cat -s myfile.txt   Отображает последовательные пустые строки как одну
$ cat -n myfile.txt   Показывает номера строк с выводимыми данными
$ cat -b myfile.txt   Отображает номера только для заполненных строк
```

Однако если блок вашего текста превышает длину нескольких строк, использование команды cat становится нецелесообразным. В этом случае для просмотра всего текста или его пролистывания вам понадобятся приложения более высокого уровня.

Для **просмотра файла с его начала** используйте команду head:

```
$ head myfile.txt
$ cat myfile.txt | head
```

В обеих командных строках для отображения первых десяти строк файла используется команда head. Для выведения любого количества строк от начала файла вы можете определить их количество в качестве параметра:

```
$ head -n 50 myfile.txt      Отображает первые 50 строк файла
$ ps auwx | head -n 15       Выводит первые 15 строк выхода команды ps
```

Выполнить это можно также с помощью следующего устаревшего (но более короткого) синтаксиса:

```
$ head -50 myfile.txt
$ ps auwx | head -15
```

Для аналогичного **просмотра конца файла** предназначена команда tail:

```
$ tail -n 15 myfile.txt      Отображает последние 15 строк файла
$ tail -15 myfile.txt        Выводит последние 15 строк файла
$ ps auwx | tail -n 15       Отображает последние 15 строк выхода команды ps
```

Команда tail также может использоваться для **непрерывного просмотра конца файла**, когда файл создается другой программой. Это очень полезно для чтения файлов журнала в режиме реального времени во время поиска и устранения неисправностей в apache, sendmail и многих других системных службах:

```
# tail -f /var/log/messages              Просматривает системные сообщения
                                         в режиме реального времени
# tail -f /var/log/maillog               Просматривает сообщения почтового
                                         сервера в режиме реального времени
# tail -f /var/log/httpd/access_log      Просматривает сообщения веб-сервера
                                         в режиме реального времени
```

Постраничный просмотр всего текста

Когда у вас есть большой фрагмент текста и вам нужно просмотреть не только его начало или конец, потребуется приложение **для постраничного просмотра текста**. Изначально системной программой UNIX, предназначенной для постраничного вывода на экран текста, была команда more:

```
$ ps auwx | more      Постранично отображает выход ps (для отображения
                      следующей страницы нажмите Пробел)
$ more myfile.txt     Постранично отображает содержимое файла
```

Однако у команды more есть некоторые ограничения. Например, в первой команде, представленной выше, more не может осуществлять прокрутку вверх. Команда less была изначально создана как более эффективная и ориентированная на пользователя, чем more. При описании команды less обычно говорят: «What is less? less is more!» (дословно: «Что такое меньше (less)? Меньше – это больше (more)»). Мы рекомендуем вам вместо команды more всегда использовать команду less.

ПРИМЕЧАНИЕ

Команда less обладает еще одним преимуществом, заслуживающим внимания. В отличие от текстовых редакторов (например, vi), она не прочитывает весь файл сразу при его загрузке, что позволяет быстро загружать и отображать содержимое больших файлов.

В приведенных ниже примерах команда less используется с тем же синтаксисом, что и more:

```
$ ps auwx | less            Постранично отображает выход команды ps
$ cat myfile.txt | less     Постранично отображает содержимое файла
$ less myfile.txt           Постранично отображает текстовый файл
```

Команда less позволяет **перемещаться по файлу** с помощью клавиш управления курсором ↑ и ↓, а также клавиш Page Up, Page Down и Пробел. Если вы используете команду less по отношению к файлу, то, чтобы открыть его в редакторе, нажмите клавишу V. Тип используемого в данном случае редактора будет определен исходя из переменных среды, заданных для вашей учетной записи. Редактор берется из переменной среды VISUAL, если она определена, или EDITOR, если редактор VISUAL не определен. Если ни один из редакторов не определен, less использует редактор JOE (в Ubuntu).

ПРИМЕЧАНИЕ

В подобном случае в других версиях Linux в качестве редактора по умолчанию задействуется vi.

Для выхода из режима просмотра файла нажмите сочетание клавиш Ctrl+C. Как и в редакторе vi, при просмотре файла с помощью команды less **поиск строки** осуществляется нажатием клавиши /, после чего требуется ввести искомую строку и нажать клавишу Enter. Для вывода остальных найденных строк последовательно нажимайте клавиши / и Enter.

Для **перемещения по тексту вперед и назад** используйте клавиши F и B соответственно. Например, десять нажатий клавиши F переместит вас на десять строк вперед, а 15 нажатий клавиши B – на 15 строк назад. Для перемещения на полови-

ну экрана вниз предназначена клавиша D, а для перемещения на половину экрана вверх — клавиша U.

азбиение текста на страницы

Команда pr предоставляет возможность быстро форматировать фрагмент текста для подготовки его к последующему выводу на печать. Она может быть особенно полезна, если требуется распечатать результаты выполнения каких-либо команд без необходимости запускать программу для редактирования или текстовый редактор. Используя команду pr, вы можете разбивать текст на части, добавляя в заголовок каждой страницы текущую дату, время, имя файла и номер страницы:

```
$ dpkg-query -l | sort | pr --column=2 | less  Разбивает список программных
                                               пакетов на два столбца
```

В данном примере команда rpm -qa отображает список всех программных пакетов, установленных в вашей системе, передает этот список команде sort, чтобы рассортировать его в алфавитном порядке, затем направляет этот список команде pr, преобразует его список в список из двух столбцов (--columns=2) и разбивает его. В результате выполняется команда less, позволяя вам просмотреть отформатированный текст.

Вы также можете **направить выводимую информацию в файл или на печать**, не просматривая ее:

```
$ dpkg-query -l | sort | pr --column=2 > pkg.txt   Направляет выход pr в файл
$ dpkg-query -l | sort | pr --column=2 | lpr       Направляет выход команды pr
                                                   на принтер
```

Другие операции с текстом, которые вы можете выполнять с помощью команды pr, включают в себя увеличение междустрочного интервала в два раза (параметр -d), отображение управляющих символов (параметр -c) и смещение текста на определенное количество знаков от левого края (например, параметр -o 5 позволяет сместить текст на пять знаков).

оиск в тексте

Команда grep очень удобна, если необходимо выполнить сложный поиск строк в файле. На самом деле выражение «to grep» уже вошло в компьютерный жаргон, как и «to Google» («гуглить») — в популярный. Вот примеры использования команды grep:

```
$ grep francois myfile.txt            Показывает строки, содержащие слово francois
# grep 404 /var/log/httpd/access_log  Показывает строки, содержащие число 404
$ ps auwx | grep init                 Показывает строки init из выхода ps
$ ps auwx | grep "\[*\]"              Показывает команды, заключенные в скобки
$ dmesg | grep "[ ]ata\|^ata"         Показывает информацию об устройстве ядра ata
```

У этих командных строк, помимо того, что они являются примерами использования команды grep, есть некоторые частные случаи применения. Выполнив поиск

числа 404 в файле access_log, вы сможете просмотреть запросы на поиск ненайденных страниц, поступивших на веб-сервер на страницы (возможно, кто-то пытается использовать вашу систему или веб-страницу, которую вы переместили или забыли создать). Отображение строк выхода команды ps, заключенных в скобки, позволяет просмотреть команды, параметры которых команда ps отобразить не может. Последняя команда проверяет защитное кольцо ядра на наличие любой информации о любых устройствах ATA, таких как жесткие диски или CD-приводы.

Команда grep способна также осуществлять **одновременный рекурсивный поиск нескольких файлов**. Следующая команда осуществляет рекурсивный поиск строки VirtualHost в каталогах /etc/httpd/conf и /etc/ httpd/conf.d:

```
$ grep -R VirtualHost /etc/httpd/conf*
```

Стоит отметить, что в каталоге /etc/httpd может не быть файлов, начинающихся с conf: это зависит от состава установленных у вас программ. Эта же техника может быть применена и к другим файлам.

Чтобы **найти конкретные строки**, в которых встречается элемент поиска, добавьте в команду grep номера строк (параметр -n):

```
$ grep -Rn VirtualHost /etc/httpd/conf*
```

Чтобы **разбить найденные данные на столбцы разного цвета**, добавьте параметр --color:

```
$ grep --color -Rn VirtualHost /etc/httpd/conf*
```

По умолчанию в многофайловом поиске имя файла отображается для каждого результата поиска. Чтобы **отключить вывод имен файлов**, используйте параметр -h. Следующий пример демонстрирует выполнение поиска строки sshd в файле auth.log:

```
$ grep -h sshd /var/log/auth.log
```

Если вы хотите сделать поиск сообщений **нечувствительным к используемому регистру**, используйте параметр -i:

```
$ grep -i selinux /var/log/messages          Ищет в файле слово selinux
                                             (независимо от регистра)
```

Для **отображения только имен включенных в поиск файлов** добавьте к команде параметр -l:

```
$ grep -Rl VirtualHost /etc/httpd/conf*
```

Для **отображения всех строк, которые не содержат указываемое сочетание**, добавьте параметр -v:

```
$ grep -v  200  /var/log/httpd/access_log*   Отображает строки,
                                             не содержащие число 200
```

ПРИМЕЧАНИЕ

При передаче выводимых данных из ps в grep для предотвращения появления процесса grep в результатах поиска применяют одну хитрость: # ps auwx | grep "[i]nit".

Определение количества элементов

Иногда необходимо знать количество строк, содержащихся в искомом фрагменте текста. Команда wc позволяет **подсчитывать количество полученных строк**. Например, следующая команда отображает количество записей конкретного IP-адреса в файле журнала Apache:

```
$ grep 192.198.1.1 /var/log/httpd/access-log | wc -l
```

Команда wc может быть использована и для других целей. По умолчанию она **отображает количество строк, слов и байт в файле**:

```
$ wc /var/log/dmesg          Отображает счетчики для одного файла
436 3847 27984 /var/log/dmesg
$ wc /var/log/*.log          Отображает отдельные значения для каждого
                             файла и общие для всех файлов
 305     3764 25772 /var/log/auth.log
 780     3517 36647 /var/log/bootstrap.log
 350     4405 39042 /var/log/daemon.log
10109 60654 669687 /var/log/dpkg.log
  71      419  4095 /var/log/fontconfig.log
 1451 19860 135252 /var/log/kern.log
   0        0     0 /var/log/lpr.log
   0        0     0 /var/log/mail.log
   0        0     0 /var/log/pycentral.log
   0        0     0 /var/log/scrollkeeper.log
  108  1610 13864 /var/log/user.log
   0        0     0 /var/log/uucp.log
  12       43   308 /var/log/wvdialconf.log
  890    6717 46110 /var/log/Xorg.0.log
14076 100989 970777 total
```

Упорядочивание выводимых данных

Вам также может понадобиться **сортировать содержимое файла или результат выполнения какой-либо команды** для упорядочивания неорганизованно выводимых данных. Следующие примеры отображают названия всех установленных на данный момент программных пакетов RPM, отбирают из них все, содержащие в названии слово kernel, и отображают полученные результаты в алфавитном порядке (прямом и обратном):

```
$ dpkg-query -l | grep kernel | sort     Отображает в прямом алфавитном порядке
$ dpkg-query -l | grep kernel | sort -r  Отображает в обратном алфавитном порядке
```

Следующая команда **сортирует процессы по уровню использования памяти** (четвертое поле выхода команды ps). Параметр -k используется для указания поля, используемого для сортировки. Запись 4,4 показывает, что именно четвертое поле (и только оно) является ключевым.

```
$ ps auwx | sort -r -k 4,4
```

Следующая команда **отображает загруженные модули ядра в порядке увеличения их размера**. Параметр n указывает команде sort на необходимость интерпретировать второе поле как номер, а не как строку:

```
$ lsmod | sort -k 2,2n
```

Поиск текста в бинарных файлах

Иногда возникает необходимость прочесть текст в кодировке ASCII, находящийся внутри бинарного файла, что позволяет узнавать много информации об исполняемом файле. В этих случаях для **извлечения любого доступного для чтения текста ASCII** используйте команду strings. Эта команда является частью программного пакета binutils и устанавливается в Ubuntu по умолчанию. Вот несколько примеров ее использования:

```
$ strings /bin/ls | grep -i libc    Определяет наличие libc в ls
$ cat /bin/ls| strings              Отображает все тексты ASCII в ls
$ strings /bin/ls                   Отображает все тексты ASCII в ls
```

Замена текста

Поиск текста в файле часто является первым шагом перед его заменой. Для редактирования потоков текста предназначена команда sed. Команда sed является полноценным языком сценариев. В примерах, которые приводятся ниже, раскрываются основные принципы замены текста с помощью команды sed.

Если вы хорошо знакомы с командами редактора vi, позволяющими заменять текст, то sed покажется вам похожим на них. Команда, приведенная в следующем примере, **заменяет только первый случай употребления** слова francois на chris **для каждой строки**. Здесь команда sed получает данные из канала и выводит их через stdout (экран монитора):

```
$ cat myfile.txt | sed s/francois/chris/
```

Если добавить в конец строки замены параметр g, как в следующей команде, то словом chris будет заменен каждый случай употребления слова francois. В следующем примере ввод даных производится из файла myfile.txt, а выход направляется в файл mynewfile.txt:

```
$ sed s/francois/chris/g < myfile.txt > mynewfile.txt
```

В следующем примере первые случаи употребления текста /home/bob заменяются текстом /home2/bob из файла /etc/passwd file (необходимо отметить, что данная команда не изменяет файл, а лишь выводит измененный текст). Она может пригодиться, например, если учетные записи пользователей были перенесены в новый каталог (или на новый диск), названный случайно home2. Здесь, чтобы избежать использования /, применяются кавычки и \, которые не являются знаками-разделителями:

```
$ sed 's/\/home\/bob/\/home2\/bob/g' < /etc/passwd
```

Хотя / является знаком-разделителем команды sed по умолчанию, вы **можете назначить вместо него любой другой символ**. Изменение знака-разделителя может упростить работу с командой, если в строке содержатся знаки /. Например, предыдущая команда, в которой содержится путь, может быть заменена любой из следующих:

```
$ sed 's-/home/bob/-/home2/bob/-' < /etc/passwd
$ sed 'sD/home/bob/D/home2/bob/D' < /etc/passwd
```

В первом примере знак дефис (-) используется в качестве знака-разделителя. Во второй строке роль разделителя выполняет буква D.

Команда sed может **одновременно выполнять несколько замен**, если перед каждой из них использовать параметр -e. Ниже, в тексте, извлеченном из файла myfile.txt, все случаи употребления francois были изменены на FRANCOIS, a chris — на CHRIS:

```
$ sed -e s/francois/FRANCOIS/g -e s/chris/CHRIS/g < myfile.txt
```

Кроме того, команда sed может использоваться для **добавления в текстовый поток символов перехода на новую строку**. При появлении в ходе выполнения команды слова Enter нажмите клавишу Enter. Символ > во второй строке сгенерирован консолью bash, его вводить не нужно.

```
$ echo aaabccc | sed 's/b/\Enter
> /'
aaa
ccc
```

Тем не менее, если вам понадобится заменить символы перехода на новую строку, воспользуйтесь командой tr.

реобразование и удаление символов

Команда tr предоставляет простой способ **выполнения быстрых замен символов в процессе работы**. В следующем примере все знаки перехода на новую строку заменяются пробелами, в результате чего все отображаемые файлы из данного каталога выводятся на одну строку:

```
$ ls | tr '\n' ' '          Заменяет символы перехода на новую строку пробелами
```

Команда tr может использоваться для **замены одного символа другим**, но, в отличие от sed, она не работает со строками. Следующая команда заменяет все прописные буквы «f» заглавными.

```
$ tr f F < myfile.txt       Заменяет каждую f в файле на F
```

Вы также можете использовать команду tr для **удаления символов**:

```
$ ls | tr -d '\n'           Удаляет знаки перехода на новую строку
                            (образуя на выходе одну строку)
$ tr -d f < myfile.txt      Удаляет из файла все буквы f
```

Когда вам необходимо **определить ряды символов для дальнейшей работы**, команда tr может помочь выполнить некоторые трюки. Ниже представлен пример изменения регистра букв с нижнего на верхний:

```
$ echo chris | tr a-z A-Z    Преобразует chris в CHRIS
CHRIS
```

Тот же результат может быть достигнут при выполнении следующей команды:

```
$ echo chris | tr '[:lower:]' '[:upper:]'      Преобразует chris в CHRIS
```

Определение различий между двумя файлами

Если у вас есть две версии одного файла, иногда может быть полезно **определить, имеют ли они различия**. Например, обновляя программный пакет, вы можете сохранить старый конфигурационный файл под новым именем (к примеру, config.old или config.bak) и таким образом оставить старую конфигурацию. В этих случаях для определения различающихся строк в старой и новой конфигурации вы можете использовать команду diff:

```
$ diff config config.old
```

Вы можете перевести выход diff в так называемый *унифицированный формат*, который легче воспринимается человеком. Он добавляет три контекстные строки до и после каждой группы измененных строк, по которым выводит отчет, а затем для отображения различий между файлами добавляет символы + и -. В следующих нескольких командах создается файл (f1.txt), содержащий последовательность цифр (1–7), и файл (f2.txt) с одной из этих цифр, измененных с помощью команды sed, после чего два этих файла сравниваются с помощью команды diff:

```
$ seq 1 7 > f1.txt              Передает последовательность цифр в файл f1.txt
$ cat f1.txt                    Отображает содержимое файла f1.txt
1
2
3
4
5
6
7
$ sed s/4/FOUR/ < f1.txt > f2.txt    Изменяет 4 на FOUR и передает результат
                                                     в файл f2.txt
$ diff f1.txt f2.txt
4c4                             Отображает измененную четвертую строку в файле
< 4
---
> FOUR
$ diff -u f1.txt f2.txt         Отображает унифицированный выход команды diff
--- f1.txt 2007-09-07 18:26:06.000000000 -0500
+++ f2.txt 2007-09-07 18:26:39.000000000 -0500
@@ -1,7 +1,7 @@
1
```

```
2
3
-4
+FOUR
5
6
7
```

В данном примере команда `diff -u` отобразила только дополнительную информацию: дату и время изменения стандартной команды `diff`. Команда `sdiff` может быть использована для выполнения еще одной задачи: она может **объединить выход двух файлов**, как показано в следующем примере:

```
$ sdiff f1.txt f2.txt
1                              1
2                              2
3                              3
4                            | FOUR
5                              5
6                              6
7                              7
```

Другим вариантом команды `diff` является команда `vimdiff`, которая одновременно открывает два файла в редакторе vim и выделяет различия между ними цветом *о каждом процессе*. Аналогичным образом команда `gvimdiff` открывает два файла в редакторе gvim.

1МЕЧАНИЕ

Для запуска программы gvim или выполнения команды gvimdiff вам необходимо установить программный пакет vim-gnome.

Выход команды `diff -u` может отображаться и через команду `patch`. Она использует старый и измененный файлы в качестве источников входящих данных, а затем возвращает измененный файл. В следующем примере команда `diff` используется для сравнения двух файлов, а затем к первому файлу применяются найденные изменения:

```
$ diff -u f1.txt f2.txt > patchfile.txt
$ patch f1.txt < patchfile.txt
patching file f1.txt
$ cat f1.txt
1
2
3
FOUR
5
6
7
```

Именно таким образом многие разработчики OSS (Operational Support System) (включая разработчиков ядра) распространяют свои заплатки. Команды `patch` и `diff` могут использоваться для целого дерева каталогов, однако эта тема уже выходит за рамки данной книги.

Использование команд awk и cut для столбцов процессов

Другим мощным инструментом редактирования текста является команда awk. Она представляет собой полноценный язык программирования. В следующих примерах рассматривается несколько хитростей, связанных только с **извлечением столбцов из текста**, хотя с помощью этой команды вы сможете сделать гораздо больше:

```
$ ps auwx | awk '{print $1,$11}'            Отображает столбцы 1 и 11 команды ps
$ ps auwx | awk '/francois/ {print $11}'    Отображает процессы
                                            пользователя francois
$ ps auwx | grep francois | awk '{print $11}' То же
```

В первом примере отображается содержимое первого (имя пользователя) и одиннадцатого столбцов (имя команды) из выходных данных текущих процессов команды ps (ps auwx). Следующие две команды выводят те же данные, но в первом случае поиск всех процессов пользователя francois осуществляется с помощью команды awk, а во втором — с помощью команды grep. В каждом их этих случаев при отображении процессов пользователя francois выводится также и 11-й столбец (имя команды).

По умолчанию команда awk в интервалах между столбцами устанавливает разделительный знак. С помощью параметра F вы можете **установить любой другой разделитель**:

```
$ awk -F: '{print $1,$5}' /etc/passwd      В качестве знака-разделителя
                                           для отображения столбцов
                                           устанавливается двоеточие
```

То же самое можно проделать и с помощью команды cut. Как и в предыдущем примере, в качестве знака-разделителя столбцов для обработки информации из файла /etc/passwd будет установлен знак двоеточия:

```
$ cut -d: -f1,5 /etc/passwd       В качестве знака-разделителя для отображения
                                  столбцов устанавливается двоеточие
```

Команда cut также может быть использована при работе **с несколькими полями значений**. Следующая команда отображает столбцы 1–5 файла /etc/passwd file:

```
$ cut -d: -f1-5 /etc/passwd       Отображает столбцы 1-5
```

Чтобы **отобразить все столбцы от заданного номера и выше**, вы можете использовать следующий прием. Следующая команда отображает все столбцы файла /etc/passwd, начиная от пятого:

```
$ cut -d: -f5- /etc/passwd        Показывает все столбцы, начиная от пятого
```

Команду awk предпочтительнее использовать, если столбцы разделяются различным количеством интервалов (в выходе команды ps), а команду cut — при работе с файлами, разделенными запятыми (,) или двоеточиями (:), как в файле /etc/password.

Конвертирование текстовых файлов в различные форматы

Используемые в UNIX символы конца строки (\n) отличаются от тех, которые используются в MS-DOS/Windows (\r\n). Эти специальные символы текстового файла можно просмотреть, воспользовавшись командой od:

```
$ od -c -t x1 myfile.txt
```

Чтобы эти символы правильно отображались после копирования из одной среды в другую, необходимо **конвертировать файлы**:

```
$ unix2dos < myunixfile.txt > mydosfile.txt
$ cat mydosfile.txt | dos2unix > myunixfile.txt
```

Продемонстрированная в первом примере команда (unix2dos) конвертирует файл открытого текста Linux или UNIX (myunixfile.txt) в текстовый формат DOS или Windows (mydosfile.txt). Команда из второго примера (dos2unix) осуществляет противоположную операцию — конвертирует файл DOS/Windows в файл Linux/UNIX. Для использования данных команд у вас должен быть установлен программный пакет tofrodos.

Резюме

В операционных системах Linux и UNIX для настройки системы, документации и возвращаемых данных, а также для многих видов хранимой информации традиционно используются файлы открытого текста. Поэтому было создано множество команд, позволяющих осуществлять поиск, редактирование и выполнение других действий над файлами открытого текста. Даже при современных интерфейсах GUI умение работать с файлами открытого текста является необходимым для тех, кто хочет стать продвинутым пользователем Linux.

В данной главе были описаны некоторые наиболее популярные команды, предназначенные для работы с файлами открытого текста Linux. Среди этих команд присутствуют команды текстовых редакторов (таких как vi, nano и JOE), а также команды, способные редактировать разделяющиеся данные (например, sed и awk). Кроме того, здесь были приведены примеры использования команд упорядочивания текста (sort), отображения статистических данных (wc) и преобразования символов в тексте (tr).

6 Использование мультимедийных данных

Если вам необходимо лишь прослушать песню или конвертировать изображение либо аудиофайл в другой формат, нет необходимости использовать приложения GUI. Для работы с мультимедиафайлами (аудиофайлы или изображения) существуют быстрые и удобные консольные команды. Если же вам нужно работать с целыми пакетами мультимедиафайлов, то для применения одного действия к нескольким файлам вы сможете использовать команду, которую используете для преобразования одного файла.

В данной главе рассматриваются консольные приложения, предназначенные для работы со звуком и цифровыми изображениями.

Звук

Для операционных систем Linux предусмотрены команды, позволяющие работать с десятками форматов аудиофайлов. Для прослушивания аудиофайлов можно использовать такие команды, как `ogg123`, `mpg321` и `play`. Кроме того, существуют команды для оцифровки музыки с музыкальных компакт-дисков и ее сжатия для последующего эффективного хранения. Существуют даже команды, позволяющие осуществлять стриминг аудио, чтобы каждый пользователь сети мог прослушивать ваш список воспроизведения.

Проигрывание музыки

В зависимости от формата воспроизводимой вами музыки вы можете выбирать тот или иной консольный проигрыватель Linux. Команда `play` (основанная на пакете `sox`, описываемом ниже) может проигрывать аудиофайлы многочисленных свободно доступных форматов. Для воспроизведения музыки популярных открытых форматов, таких как Ogg Vorbis, Free Lossless Audio Codec (FLAC) и Speex, вы можете использовать проигрыватель `ogg123`. Проигрыватель же `mpg321`, доступный

через сторонние интернет-репозитории, является популярным проигрывателем музыкальных файлов в формате MP3.

Для использования команды play необходим программный пакет sox, который можно установить с помощью следующей команды:

```
$ sudo apt-get install sox
```

Чтобы **просмотреть список доступных для воспроизведения аудиоформатов и эффектов**, выполните команду sox -h:

```
$ sox -h
  ...
Supported file formats: 8svx aif aifc aiff aiffc al alsa au auto avr cdda cdr cvs
cvsd dat dvms fssd gsm hcom ima ircam la lu maud nist nul null ogg ossdsp prc raw
s3 sb sf sl smp snd sndt sou sph sw txw u3 u4 ub ul uw vms voc vorbis vox wav wve
xa
Supported effects: allpass band bandpass bandreject bass chorus compand dcshift
deemph dither earwax echo echos equalizer fade filter flanger highpass lowpass
mcompand mixer noiseprof noisered pad pan phaser pitch polyphase repeat resample
reverb reverse silence speed stat stretch swap synth treble tremolo trim vibro vol
```

Для воспроизведения звуков команда play использует кодировку sox. Ниже приведено несколько примеров использования команды play для воспроизведения музыкальных файлов:

```
$ play inconceivable.wav     Проигрывает файл WAV (извлеченный
                             из компакт-диска в том числе)
$ play *.wav                 Проигрывает файлы WAV из каталога (до 32 штук)
$ play hi.au vol .6          Проигрывает файл AU, уменьшает громкость
                             (снижает уровень искажений)
$ play -r 14000 short.aiff   Проигрывает файл AIFF с частотой дискретизации
                             14 000 Гц
```

Для проигрывания файлов Ogg Vorbis установите программный пакет vorbis-tools. Ниже приведены примеры воспроизведения OGG-файлов (**www.vorbis.com**) с помощью ogg123:

```
$ ogg123 mysong.ogg                 Проигрывает OGG-файл
$ ogg123 /usr/share/example-content/ubuntu\ Sax.ogg  Проигрывает файл-образец
$ ogg123 http://vorbis.com/music/Lumme-Badloop.ogg   Проигрывает файл, размещенный
                                                     по указанному интернет-адресу
$ ogg123 -z *.ogg                   Проигрывает файлы в случайном порядке
$ ogg123 /var/music/                Проигрывает файлы из каталога /var/music
                                    и подкаталога dirs
$ ogg123 -@ myplaylist              Проигрывает песни из списка воспроизведения
```

Список воспроизведения является обычным списком каталогов или отдельных OGG-файлов, которые необходимо воспроизвести. Если каталог занесен в список, то воспроизводятся все OGG-файлы, внесенные в него и все его подкаталоги. При воспроизведении нескольких файлов, чтобы **пропустить песню**, нажмите сочетание клавиш Ctrl+C. **Для выхода** дважды нажмите сочетание Ctrl+C.

Для использования проигрывателя mpg321 необходимо установить программный пакет mpg321. Ниже приведено несколько примеров воспроизведения аудио в формате MP3 с помощью проигрывателя mpg321:

```
$ mpg321 yoursong.mp3          Проигрывает MP3-файл
$ mpg321 -@ mp3list            Проигрывает аудиофайлы из списка воспроизведения
$ cat mp3list | mpg321 -@      Передает список воспроизведения в проигрыватель
                               mpg321
$ mpg321 -z *.mp3              Проигрывает файлы в случайном порядке
$ mpg321 -Z *.mp3              То же, что и с -z, но аудиофайлы воспроизводятся
                               бесконечно
```

Список воспроизведения mpg321 является обычным списком файлов. Направив выход команды ls в файл, вы можете создать список воспроизведения. Если вы не планируете использовать список из мест, к которым требуется указывать относительный путь, то указывайте полный путь к файлам.

Управление уровнем звука

Вид используемых для активизации аудиоустройств и управления уровнем звука консольных аудиоприложений зависит от типа используемой вами звуковой системы. Звуковая система Advanced Linux Sound Architecture (ALSA) («улучшенная звуковая архитектура Linux») является системой, используемой сегодня в большинстве операционных систем Linux. Звуковая система Open Source Sound System (OSS) («звуковая система Open Source») функционирует уже давно и используется сейчас только с более старым оборудованием. В общем, для управления уровнем звука при использовании ALSA предназначена команда alsamixer, а при использовании OSS — aumix.

ALSA является звуковой системой по умолчанию для большинства операционных систем Linux. Если добавить загружаемые модули, активизирующие интерфейсы устройства OSS, то аудиоприложения, для работы с которыми необходим интерфейс устройства OSS, будут также работать и с ALSA. Чтобы проверить, **загружены ли модули OSS**, такие как snd-pcm-oss (имитирует /dev/dsp и /dev/audio), snd-mixer-oss (имитирует /dev/mixer) и snd-seq-oss (имитирует /dev/sequencer), выполните следующую команду:

```
# lsmod | grep snd
```

Если модули загружены, то для управления уровнем звука приложений OSS может использоваться **проигрыватель alsamixer**, который можно **запустить** следующим образом:

```
$ alsamixer                  Отображает экран alsamixer и отражает процесс
                             воспроизведения
$ alsamixer -V playback      Отображает только каналы воспроизведения
                             (установленные по умолчанию)
$ alsamixer -V all           Отображает процесс воспроизведения и вид ввода
$ alsamixer -c 1             Использует alsamixer на второй (1) звуковой карте
```

Полоски, отражающие уровень звука, доступны для каждого звукового канала. Для **выделения отдельных каналов** (Master (главный), PCM, Headphone (науш-

ники) и др.) используйте клавиши управления курсором → и ←. Для увеличения и уменьшения уровня звука каждого из каналов используйте клавиши ↑ и ↓. Выделив канал, нажмите клавишу M для **выключения или включения воспроизведения** этого канала. Чтобы **назначить канал в качестве входящего** (для записи с аудиовхода), нажмите Пробел при выделенном входном канале (Mic (микрофон), Line (линейный вход) и т. д.). Для выхода из alsamixer нажмите сочетание клавиш Alt+Q или клавишу Esc. Для переключения между настройками воспроизведения, ввода и общим видом используйте клавишу Tab.

Приложение микширования звука aumix (для использования которого необходимо установить программный пакет aumix) может работать в режиме, ориентированном на работу с экраном или в командном режиме. В текстовом режиме **для изменения или отображения настроек** вы можете использовать параметры соответствующей команды:

```
$ aumix -                  Отображает уровень левого и правого каналов
$ aumix -l q -m q          Отображает текущие настройки только для линейного
                           и микрофонного входов
$ aumix -v 80 -m 0         Устанавливает уровень звука на 70 %, а уровень
                           микрофонного входа – на 0
$ aumix -m 80 -m R -m q    Устанавливает уровень микрофонного входа на 80 %,
                           активизирует его для записи и отображает его сигнал
$ aumix                    Если не использовать параметры, aumix запускается
                           в экранном режиме
```

В экранном режиме приложение aumix отображает все доступные аудиоканалы. В этом режиме **для изменения отображаемых настроек звука используйте клавиши на клавиатуре**. Для выбора каналов используйте клавиши Page Up, Page Down и клавиши управления курсором ↑ и ↓. Для изменения громкости используйте клавиши ← и →. Для выключения звука текущего канала нажмите клавишу M. Чтобы выбрать текущий канал в качестве источника записи, нажмите Пробел. Используя мышь, вы можете управлять уровнем звука, балансом и текущим каналом записи.

Оцифровка музыки с компакт-дисков

Чтобы вы могли проигрывать свою музыкальную коллекцию с жесткого диска, в Linux существуют различные приложения, позволяющие конвертировать музыку, записанную на компакт-диски, в WAV-файлы, а затем копировать ее на жесткий диск. После этого, используя такие приложения, как oggenc (Ogg Vorbis), flac (FLAC) или lame (MP3), оцифрованные файлы можно сжимать для уменьшения занимаемого на диске места.

ПРИМЕЧАНИЕ

Для оцифровки и сжатия аудиоданных, записанных на компакт-диски, существуют прекрасные графические приложения, такие как grip и sound-juicer. Поскольку эти приложения поддерживают CDDB, они, чтобы отображать выходные данные (имя исполнителя, название альбома, песни и т. д.), также могут использовать информацию о музыке, записанной на компакт-диск. Однако в данном разделе описывается только использование этих команд для ручной оцифровки и сжатия музыкальных компакт-дисков.

С помощью команды cdparanoia вы можете проверить, способен ли ваш CD-привод оцифровывать Compact Disc Digital Audio (CDDA), извлекать аудиодорожки и копировать их на ваш жесткий диск. Для этого вставьте музыкальный компакт-диск в привод и выполните следующую команду:

```
$ cdparanoia -vsQ
   ...
Checking /dev/cdrom for cdrom...
Checking for SCSI emulation...
Checking for MMC style command set...
Verifying CDDA command set...
   ...
Table of contents (audio tracks only):
track          length               begin        copy pre ch
============================================================
  1.     18295 [04:03.70]   0 [00:00.00]        no no 2
  2.     16872 [03:44.72]   18295 [04:03.70]    no no 2
   ...
 11.     17908 [03:58.58]   174587 [38:47.62]   no no 2
 12.     17342 [03:51.17]   192495 [42:46.45]   no no 2
TOTAL 209837 [46:37.62]     (audio only)
```

Сокращенный формат вывода данных позволяет увидеть, что команда cdparanoia проверяет возможности /dev/cdrom, выполняя поиск эмуляторов SCSI и поддержки набора команд MMC, а также проверяет, может ли данный привод обрабатывать информацию CDDA. Только после этого выводится информация о каждой дорожке. Ниже приведены примеры команд cdparanoia, использующихся для **оцифровки компакт-дисков и записи их на жесткий диск**:

```
$ cdparanoia -B                Конвертирует дорожки в WAV-файлы
$ cdparanoia -B - "5-7"        Конвертирует дорожки 5-7 в раздельные файлы
$ cdparanoia -- "3-8" abc.wav  Конвертирует дорожки 3-8 в один файл (abc.wav)
$ cdparanoia - "1:[40]-"       Конвертирует файлы, начиная с 40 секунды
                               первой дорожки и до конца компакт-диска
$ cdparanoia -f -- "3"         Конвертирует дорожку 3 и сохраняет ее
                               в формате AIFF
$ cdparanoia -a - "5"          Конвертирует дорожку 5 и сохраняет ее
                               в формате AIFF
$ cdparanoia -w - "1" my.wav   Конвертирует первую дорожку и называет ее my.wav
```

Конвертирование музыки

Следующим шагом после записи музыкального файла на жесткий диск обычно является его сжатие. Среди популярных конвертеров можно назвать oggenc, flac и lame, конвертирующие файлы в форматы Ogg Vorbis, FLAC и MP3 соответственно.

С помощью oggenc вы можете работать с исходным форматом или с аудиофайлами или потоками аудиоданных в форматах WAV, AIFF, FLAC, а также конвертировать их затем в формат OGG. Хотя формат OGG допускает большие потери качества, сжатие WAV-файлов позволяет добиваться очень хорошего качества,

несмотря на то что размер файла уменьшается приблизительно в десять раз. Вот несколько примеров использования команды oggenc:

```
$ oggenc ab.wav                    Конвертирует WAV в OGG (ab.ogg)
$ oggenc ab.flac -o new.ogg        Конвертирует FLAC вOgg (new.ogg)
$ oggenc ab.wav -q 9               Повышает качество сжатия до 9
```

По умолчанию качество (параметр -q) сжатия командой oggenc установлено на уровень 3, однако вы можете **устанавливать любое его значение**: от -1 до 10 (включая дробные числа, например 5.5).

```
$ oggenc NewSong.wav -o NewSong.ogg \
    -a Bernstein -G Classical \
    -d 06/15/1972 -t "Simple Song" \
    -l "Bernsteins Mass" \
 -c info="From Kennedy Center"
```

Показанная выше команда конвертирует песню MySong.wav в MySong.ogg. Имя исполнителя Bernstein, а стиль музыки Classical, дата написания композиции определена как 15 июня 1972 года, композиция называется Simple Song, а альбом — Bernsteins Mass, комментарий — From Kennedy Center. Если вы вводите всю команду в одну строку, то обратные слэши не нужны, однако если вы все же добавите их, убедитесь, что после них нет пробелов.

В предыдущем примере в **заголовок конечного OGG-файла добавляется некоторая информация**, просмотреть которую можно, воспользовавшись командой ogginfo:

```
$ ogginfo NewSong.ogg
Processing file "NewSong.ogg"...
   ...
Channels: 2
Rate: 44100
Nominal bitrate: 112.000000 kb/s
User comments section follows...
   info=From Kennedy Center
   title=Simple Song
   artist=Bernstein
   genre=Classical
   date=06/15/1972
   album=Bernsteins Mass
Vorbis stream 1:
   Total data length: 3039484 bytes
   Playback length: 3m:25.240s
   Average bitrate: 118.475307 kb/s
Logical stream 1 ended
```

Здесь вы можете видеть, что в процессе сжатия были добавлены комментарии. Параметр -c использовался для установки произвольно выбранного поля (в данном случае info) с некоторым значением заголовка. Помимо комментариев, у этого файла присутствуют два канала, и он был записан при частоте дискретизации 44 100 Гц. Из полученной информации можно узнать объем записанных данных, продолжительность записи и среднюю частоту дискретизации.

Команда flac похожа на oggenc, отличие состоит лишь в том, что файлы WAV, AIFF, RAW, FLAC или OGG конвертируются в формат FLAC. Поскольку flac является бесплатным аудиокодеком, не допускающим потерь качества, он является популярным методом сжатия у тех, кто хочет сохранить свободное место на диске, но в то же время ценит высокое качество звука. Если использовать значения по умолчанию, то конвертирование файлов из формата WAV в формат FLAC уменьшит их размер вдвое, что существенно отличается от пропорции 1:10 при использовании oggenc. Для использования команды flac необходимо установить программный пакет flac:

```
$ flac now.wav                  Конвертирует WAV в FLAC (now.flac)
$ sox now.wav now.aiff          Конвертирует WAV в AIFF (now.aiff)
$ flac now.aiff -o now2.flac    Конвертирует AIFF в FLAC (now.flac)
$ flac -8 top.wav -o top.flac   Увеличивает уровень сжатия до -8
```

Уровень сжатия по умолчанию равен -5. Кроме него можно также использовать уровни от 0 до -8, где наибольшее число дает наивысший уровень сжатия, а самое низкое — наилучшее качество. Чтобы конвертировать файлы в формат MP3, используя команду lame, вам нужно сначала установить программный пакет lame. Вот несколько примеров сжатия файлов в форматах WAV и AIFF с помощью команды lame:

```
$ lame in.wav                          Конвертирует файл формата WAV в MP3
                                       (in.wav.mp3)
$ lame in.wav --preset standard        Конвертирует файл в формат MP3,
                                       используя стандартные настройки
$ lame tune.aiff -o tune.mp3           Конвертирует файл формата AIFF в MP3
                                       (tune.mp3)
$ lame -h -b 64 -m m in.wav out.mp3    Устанавливает высокое качество, 64-bit,
                                       режим моно
$ lame -q 0 in.wav -o abcHQ.mp3        Конвертирует файл со значением
                                       качества, равным 0
```

Используя команду lame, вы можете устанавливать значения качества от 0 до 9 (5 является значением по умолчанию). При установке качества, равного 0, используются наилучшие алгоритмы сжатия; при использовании же значения 9 большинство алгоритмов сжатия отключается (но процесс сжатия проходит быстрее). Как и при использовании oggenc, команда lame позволяет добавлять в MP3-файл теги, которые могут использоваться позже при воспроизведении файла:

```
$ lame NewSong.wav NewSong.mp3 \
--ta Bernstein --tg Classical \
--ty 1972 --tt "Simple Song" \
--tl "Bernsteins Mass" \
--tc "From Kennedy Center"
```

Как и в рассмотренном ранее примере конвертирования файла из формата WAV в формат OGG, приведенная команда конвертирует файл MySong.wav в MySong.mp3. Имя исполнителя по-прежнему Bernstein, стиль музыки Classical, год 1972, песня называется Simple Song, а альбом — Bernsteins Mass, комментарий — From Kennedy Center.

Если вы вводите всю команду в одну строку, то обратные слэши не нужны, однако если вы все же добавите обратные слэши, убедитесь, что после них нет пробелов.

Информация из тегов отражается на экране в случае использования графических оболочек MP3-проигрывателей (например, Rhythmbox и Totem), когда они проигрывают музыку в формате MP3. Кроме того, информацию, содержащуюся в тегах, можно просматривать и при использовании консольных проигрывателей, например mpg321:

```
$ mpg321 NewSong.mp3
High Performance MPEG 1.0/2.0/2.5 Audio Player for Layer 1, 2, and 3.
   ...
Title :                                   Simple Song Artist: Bernstein
Album :                                   Bernsteins Mass Year : 1972
Comment:                                  From Kennedy Center Genre : Classical
Playing MPEG stream from NewSong.mp3 ...
MPEG 1.0 layer III, 128 kbit/s, 44100 Hz joint-stereo
```

Стриминг музыки

Если вы храните свою музыку на одном компьютере, но работаете с другим, то **настройка музыкального сервера для стриминга** может стать наиболее быстрым способом предоставить вашу музыку для прослушивания с одного или более компьютеров, находящихся в сети. Стриминг-сервер icecast и клиент аудиоисточника ices можно установить на Ubuntu, выполнив следующую команду:

```
$ sudo apt-get install icecast2 ices2
```

Ниже представлен быстрый алгоритм настройки сервера icecast и клиента ices для стриминга музыки, который необходимо выполнить на том компьютере, на котором хранятся аудиозаписи.

1. Чтобы изменить все отображаемые пароли, измените файл /etc/icecast2/icecast.xml. Для получения текущих паролей найдите поле hackme. Скорее всего, вам понадобится установить дополнительные пароли для пользователей и администратора, особенно если вы планируете открыть для других возможность предоставлять свою музыку серверу. Запомните установленные пароли, чтобы иметь возможность впоследствии их использовать. Кроме того, с помощью редактирования данного файла вы можете изменить и другие настройки, например имя хост-системы:

```
$ sudo vi /etc/icecast2/icecast.xml
```

2. Если у вас есть сетевой экран, проверьте, доступен ли его порт TCP 8000.
3. Запустите сервер icecast2 от имени суперпользователя, выполнив указанную ниже команду (на самом же деле сервер будет запущен от имени пользователя icecast2), и проверьте ее выполнение с помощью команды netstat:

```
$ sudo /etc/init.d/icecast2 start
$ sudo netstat -topnavel | grep 8000
tcp    0    0 0.0.0.0:8000      0.0.0.0:*            LISTEN
111    35790   21494/icecast    off (0.00/0/0)
```

4. Создайте каталоги, необходимые программе ices2, предоставляющей список воспроизведения и музыку серверу icecast2:

```
$ sudo mkdir /var/log/ices
$ sudo mkdir /etc/ices2
$ sudo mkdir /etc/ices2/music
```

5. Создайте список воспроизведения, воспользовавшись любым текстовым редактором или направив свой список музыки в файл. Например, если вся ваша музыка в формате OGG хранится в подкаталогах /var/music, выполните следующую команду:

```
$ find /var/music -name *.ogg > playlist.txt
```

6. Убедившись, что файл playlist.txt содержит полные пути к каждому музыкальному файлу, а сами файлы доступны серверу icecast2, скопируйте файл со списком воспроизведения в каталог /etc/ices2:

```
$ sudo cp playlist.txt /etc/ices2
```

Для редактирования файла со списком воспроизведения (удаляя или добавляя из него файлы или каталоги) можно использовать любой текстовый редактор.

7. От имени суперпользователя отредактируйте файл /etc/ices2/ices-playlist.xml, чтобы проигрывать музыку из списка воспроизведения и направлять ее на активный сервер icecast2. Начните с редактирования конфигурационного файла:

```
$ sudo cp /usr/share/doc/ices2/examples/ices-playlist.xml /etc/ices2
$ sudo vi /etc/ices2/ices-playlist.xml
```

8. В отдельных случаях вам может понадобиться изменить модули метаданных, входящих данных или модули instance (в приведенном ниже примере необходимо заменить /etc/ices2/playlist.txt на путь к вашему файлу playlist.txt):

```
<metadata>
        <name>My Music Server</name>
        <genre>Different music styles</genre>
        <description>Mix of my personal music</description>
</metadata>
<input>
        <module>playlist</module>
        <param name="type">basic</param>
        <param name="file">/etc/ices2/playlist.txt</param>
        <!-- random play -->
        <param name="random">1</param>
                        ...
</input>
<instance>
        <hostname>localhost</hostname>
        <port>8000</port>
        <password>MIcePw</password>
        <mount>/mymusic.ogg</mount>
                        ...
</instance>
```

Из показанных выше значений (выделенных полужирным шрифтом) самым критичным является адрес вашего списка воспроизведения и информация об экземпляре сервера icecast2. Пароль к серверу должен соответствовать паролю, добавленному в файл /etc/icecast2/icecast.xml file.

9. Выполнив следующую команду, запустите передачу аудиоданных клиенту ices:

```
$ sudo ices2 /etc/ices2/ices-playlist.xml &
```

10. Чтобы проверить, можете ли вы проигрывать музыку с локального компьютера, выполните следующую команду:

```
$ ogg123 http://localhost:8000/mymusic.ogg
```

11. Если тест будет завершен успешно, попробуйте воспроизвести поток icecast2 с другого компьютера вашей сети, изменив localhost на IP-адрес или имя хост-машины.
12. Если же при проведении теста возникнут проблемы, проверьте файлы журнала /var/log/icecast2 и/var/log/ices, а также пароли и адреса конфигурационных файлов.
13. После завершения настройки удалите сервис icecast2:

```
$ sudo /etc/init.d/icecast2 stop
```

Во время активности серверов icecast и ices любой компьютер, подключенный к серверу, должен получать транслируемый поток аудиоданных. Для прослушивания музыки может использоваться любой музыкальный проигрыватель, способный воспроизводить музыку с определенного HTTP-адреса (ogg123, Rhythmbox, XMMS и т. д.). Музыкальные проигрыватели Windows, поддерживающие воспроизведение транслируемых форматов, также должны работать.

ПРИМЕЧАНИЕ

Чтобы пропустить проигрываемую песню, выполните на сервере следующую команду: killall -HUP ices.

Конвертирование аудиофайлов

Утилита sox является гибким приложением, позволяющим работать с аудиофайлами свободно распространяемых форматов. Ниже представлено несколько примеров, демонстрирующих выполнение определенных задач с помощью sox.

Данная команда **объединяет два WAV-файла в один**:

```
$ sox head.wav tail.wav output.wav
```

Следующая команда **микширует два WAV-файла**:

```
$ soxmix sound1.wav sound2.wav output.wav
```

Для **отображения информации о файле** с помощью sox используйте эффект stat:

```
$ sox sound1.wav -e stat
Samples read:              208512
```

```
Length (seconds):           9.456327
Scaled by:                  2147483647.0
Maximum amplitude:          0.200592
Minimum amplitude:         -0.224701
Midline amplitude:         -0.012054
Mean norm:                  0.030373
Mean amplitude:             0.000054
RMS amplitude:              0.040391
Maximum delta:              0.060852
Minimum delta:              0.000000
Mean delta:                 0.006643
RMS delta:                  0.009028
Rough frequency:            784
Volume adjustment:          4.450
```

Чтобы **удалить звук из аудиофайла**, используйте команду `trim`:

```
$ sox sound1.wav output.wav trim 4      Удаляет четыре первые секунды
                                        аудиозаписи
$ sox sound1.wav output.wav trim 2 6    Удаляет аудиозапись до второй
                                        и после шестой секунд
```

В первом примере удаляются первые четыре секунды из файла `sound1.wav`, а результат сохраняется в файл `output.wav`. Во втором примере остается часть файла `sound1.wav` со второй по шестую секунды, остальное удаляется, а полученный файл сохраняется под именем `output.wav`.

Преобразование изображений

Если у вас есть каталоги, хранящие коллекции цифровых изображений, то возможность управлять ими из командной строки может существенно сэкономить время. Программный пакет ImageMagick поставляется с некоторыми приложениями, позволяющими преобразовывать цифровые изображения в формы, удобные для дальнейшей работы (чтобы установить данный пакет на Ubuntu, выполните команду `apt-get install imagemagick`). В этом разделе описаны некоторые команды, предназначенные для работы с цифровыми изображениями, а также приводятся примеры простых командных строк для пакетной обработки.

Получение информации об изображениях

Чтобы **получить информацию об изображении**, используйте команду `identify`:

```
$ identify p2090142.jpg
p2090142.jpg JPEG 2048x1536+0+0 DirectClass 8-bit 402.037kb
$ identify -verbose p2090142.jpg | less
 Standard deviation: 61.1665 (0.239869)
 Colors: 205713
 Rendering intent: Undefined
```

```
Resolution: 72x72
Units: PixelsPerInch
Filesize: 402.037kb
Interlace: None
Background color: white
Border color: rgb(223,223,223)
Matte color: grey74
Transparent color: black
Page geometry: 2048x1536+0+0
Compression: JPEG
Quality: 44
  ...
```

Первая команда в данном примере отображает основную информацию об изображении (имя файла, его формат, размер, класс и глубину канала). Вторая команда отображает всю информацию, которую можно извлечь из изображения. Помимо приведенной в примере информации, команда также позволяет просмотреть время создания файла, тип использованной камеры, значение диафрагмы и ISO.

Конвертирование изображений

Команда convert является наиболее качественным инструментом конвертирования изображений. Ниже приведены некоторые способы **конвертирования изображений из одного формата в другой** с помощью команды convert:

```
$ convert tree.jpg tree.png       Конвертирует файл JPEG в PNG
$ convert icon.gif icon.bmp       Конвертирует файл GIF в BMP
$ convert photo.tiff photo.pcx    Конвертирует файл TIFF в PCX
```

Среди форматов изображений, поддерживаемых командой convert, можно назвать JPG, BMP, PCX, GIF, PNG, TIFF, XPM и XWD. Ниже приведены некоторые примеры конвертирования изображений **с изменением их размера**:

```
$ convert -resize 1024x768 hat.jpg hat-sm.jpg
$ convert -sample 50%x50% dog.jpg dog-half.jpg
```

В первом примере создается изображение (файл hat-sm.jpg) с разрешением 1024 × 768 пикселов. Во втором примере изображение dog.jpg уменьшается наполовину (50% x 50%) и сохраняется под именем dog-half.jpg.

Кроме того, вы можете **поворачивать изображения** на угол от 0 до 360 °:

```
$ convert -rotate 270 sky.jpg sky-final.jpg     Разворачивает изображение на 270 °
$ convert -rotate 90 house.jpg house-final.jpg Разворачивает изображение на 90 °
```

Используя параметр -draw, вы можете **добавлять к изображению текст**:

```
$ convert -fill black -pointsize 60 -font helvetica \
   -draw 'text 10,80 "Copyright NegusNet Inc."' \
     p10.jpg p10-cp.jpg
```

В данном примере к изображению была добавлена информация об авторских правах, причем для создания надписи на изображении использовался шрифт Helvetica с размером кегля 60. Текст помещается на 10 пикселов правее и на 80 пикселов ниже верхнего левого угла. Полученное изображение было сохранено под именем p10-cp.jpg, чтобы обозначить, что в новом изображении добавлена информация об авторских правах.

Далее показано несколько интересных способов **создания эскизов** с помощью команды convert:

```
$ convert -thumbnail 120x120 a.jpg a-a.png
$ convert -thumbnail 120x120 -border 8 a.jpg a-b.png
$ convert -thumbnail 120x120 -border 8 -rotate 8 a.jpg a-c.png
```

Во всех трех примерах создаются эскизы с разрешением 120 × 120 пикселов. Во втором случае был добавлен параметр -border, позволяющий создавать рамку вокруг эскиза, делая его похожим на снимок Polaroid. В последнем примере изображение также поворачивается. На рис. 6.1 продемонстрирован результат выполнения всех трех команд.

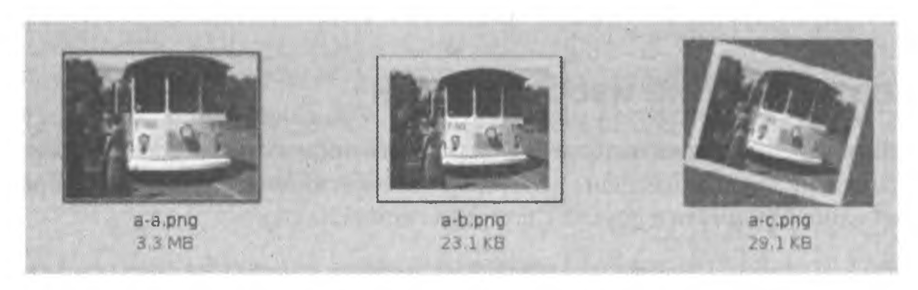


Рис. 6.1. Использование команды convert для создания эскиза, добавления рамки и переворота изображения

Кроме того, существуют способы применять к изображениям **различные эффекты**:

```
$ convert -sepia-tone 75% house.jpg oldhouse.png
$ convert -charcoal 5 house.jpg char-house.png
$ convert -colorize 175 house.jpg color-house.png
```

Параметр -sepia-tone позволяет создать эффект старой фотографии. Использование параметра -charcoal создает впечатление, что изображение было нарисовано от руки углем. При использовании же параметра -colorize цвет каждого пиксела изображения изменяется в соответствии с заданным числом (в приведенном случае 175). На рис. 6.2 в верхнем левом углу показано оригинальное изображение дома, в верхнем правом — то же изображение, но с применением эффекта sepia-tone, в нижнем левом — с применением эффекта charcoal, а изображение с применением настроек цвета показано в нижнем правом углу.

Если вы захотите увидеть еще один пример обработки изображений, попробуйте применить к изображению эффект спирали:

```
$ convert -swirl 300 photo.pcx weird.pcx
```



Рис. 6.2. Оригинальное изображение, а также с использованием эффектов sepia-tone, charcoal и colorize

Конвертирование пакетов изображений

Большинство описанных в данной главе преобразований могут быть выполнены в приложениях по работе с изображениями, такими как GIMP. Однако в полной мере использовать возможности команды convert можно только при работе через консоль. В этом случае вы можете применять одну (или все) вышеуказанные операции одновременно к целому каталогу файлов.

Например, вам может понадобиться создать эскизы для всей коллекции изображений или, возможно, уменьшить размер всех свадебных фотографий таким образом, чтобы они могли уместиться на цифровой рамке. Вам может даже понадобиться добавить информацию об авторских правах на каждое изображение, хранящееся в каталоге, прежде чем предоставлять их для общего просмотра в Интернет. Все эти действия можно осуществить с помощью уже описанных параметров команды convert, а также некоторых простых команд командного процессора.

Ниже приведен пример команды, позволяющей **изменить разрешение всех изображений каталога** на 1024 × 768 пикселов, чтобы вместить их в цифровую рамку:

```
$ cd $HOME/myimages
$ mkdir small
$ for pic in `ls *.png'
do
 echo "converting $pic"
 convert -resize 1024x768 $pic small/sm-$pic
done
```

Перед выполнением кода программа переходит в каталог `$HOME/myimages` (в котором хранятся изображения с высоким разрешением). Затем для помещения в него уменьшенных изображений создается подкаталог `small`. Сам код начинается с петли `for`, которая отображает каждый файл в текущем каталоге, имеющий разрешение PNG (вам может понадобиться указать расширение JPG или другое). После этого разрешение каждого файла меняется на 1024 × 768 пикселов и полученные файлы копируются в каталог `small`, причем к каждому имени файла добавляется приставка `sm-`.

С этим же кодом вы можете использовать любые описанные ранее строки команды `convert` или же создавать свои собственные, удовлетворяющие вашим личным пожеланиям. С его же помощью вы можете конвертировать целые каталоги с файлами изображений, и это займет у вас лишь несколько минут вместо нескольких часов работы в приложении GUI.

Резюме

Консоль может предоставить быстрые и эффективные способы работы с аудиофайлами и цифровыми изображениями. В данной главе описаны различные консольные способы воспроизведения, оцифровки, сжатия, конвертирования и стриминга аудиофайлов. Что же касается цифровых изображений, то здесь представлено множество примеров использования команды `convert` для изменения их размера, разворота, конвертирования, создания записи и выполнения других операций с ними.

7 Администрирование файловых систем

Файловые системы представляют собой структуры, посредством которых можно получать доступ к файлам, каталогам, устройствам и другим элементам операционной системы Linux. Linux поддерживает много различных типов файловых систем (ext3, VFAT, ISO9660, NTFS и т. д.), а также множество типов носителей, на которых эти файловые системы могут существовать (жесткие диски, компакт-диски, USB-носители, ZIP-дискеты и т. д.).

Создание и управление разделами дисков и файловыми системами на этих носителях является одной из важнейших задач при администрировании операционной системы Linux. Именно поэтому если вы повредите свою файловую систему, то, скорее всего, потеряете критически важные данные, хранящиеся на вашем жестком диске или съемных носителях.

В данной главе описываются команды, позволяющие разбивать дисковое пространство носителя с хранящейся на нем информацией, создавать файловые системы, монтировать и демонтировать разделы, а также проверять файловые системы на наличие ошибок и свободного пространства.

Введение в основы файловых систем

Несмотря на то что Linux поддерживает большое количество типов файловых систем, тех, на которые вы сможете установить операционную систему Linux, не так много. Для установки стандартной системы Linux на вашем жестком диске должно присутствовать только три раздела: раздел подкачки (используется для предотвращения переполнения оперативной памяти информацией), загрузочный диск, на котором хранятся загрузчик и ядро, и раздел корневой файловой системы. В разделах с загрузочной и корневой файловыми системами обычно используется файловая система типа ext3.

Файловая система ext3 основывается на файловой системе ext2, к которой была добавлена функция *журналирования*. Журналирование может упростить процесс обеспечения сохранности данных и их восстановления, в частности после некорректного завершения работы системы. Во время следующей после некорректного завершения работы системы перезагрузки не потребуется выполнения длительных проверок файловых систем, поскольку изменения, произошедшие со времени последней записи на диск, сохраняются и могут легко быть восстановлены.

В большинстве примеров данной главы, чтобы продемонстрировать процесс создания и управления файловой системой, используется файловая система ext3, однако иногда вам может понадобиться использовать и другие типы файловых систем. В табл. 7.1 приведен список различных типов файловых систем и их описания на случай, если вам понадобится их использовать.

Таблица 7.1. Типы файловых систем, поддерживаемые Linux

Тип файловой системы	Описание
ext3	Наиболее распространенная файловая система Linux, предоставляющая возможность журналирования с целью сохранения данных и возможность быстро перезагружать систему после некорректного завершения ее работы
ext2	Файловая система, предшествующая ext3, не поддерживающая функцию журналирования
iso9660	Развилась из файловой системы High Sierra (была стандартом для компакт-дисков) и может содержать расширения Rock Ridge для обеспечения возможности использования длинных имен файлов и других функций (полномочия доступа к файлу, права собственности и ссылки)
Jffs2	Журналируемая файловая система для флэш-носителей, созданная для эффективной работы с USB-накопителями. Продолжательница системы JFFS
jfs	Файловая система, используемая на IBM в OS/2 Warp и настроенная на работу с большими файловыми системами и средами с высокой производительностью
msdos	Может использоваться для монтирования файловых систем на устаревшие носители MS-DOS, например старые дискеты
ntfs	Может быть полезна, когда есть необходимость передавать файлы в файловые системы Windows (с двойной загрузкой или на сменных носителях)
reiserfs	Журналируемая файловая система, иногда используемая по умолчанию на SUSE, Slackware и других операционных системах Linux, но не поддерживаемая в Ubuntu
squashfs	Запакованная файловая система, предназначенная только для чтения данных и используемая на многих Live CD Linux
swap	Используется на разделах подкачки для временного хранения данных, когда оперативная память временно недоступна
ufs	Популярная файловая система операционных систем Solaris и SunOS от корпорации Sun Microsystems
vfat	Расширенная файловая система FAT (VFAT), которая полезна, если файловым системам нужно обмениваться файлами с более старыми системами Windows (с двойной загрузкой или на сменных носителях)
xfs	Журналируемая файловая система для среды с высокой производительностью, размер которой может расширяться до многих терабайт данных, а скорость передачи данных в которой исчисляется гигабайтами в секунду

Помимо файловых систем, рассмотренных в таблице, существуют также сетевые файловые системы, называемые *общими сетевыми* файловыми системами. Общая сетевая файловая система должна относиться к типу ext3, ntfs или другому стандартному типу, любая часть которой может быть предоставлена для общего

доступа посредством сетевых протоколов, таких как Samba (тип файловой системы smbfs или cifs), NFS (nfs) и NetWare (ncpfs).

Многие из доступных типов файловых систем либо неспособны создавать новые файловые системы, либо не поддерживаются во всех версиях Linux. Например, такие файловые системы, как minix (для операционных систем Minix), befs (для операционных систем BeOS) и affs (для операционных систем Amiga), очень полезны, когда необходимо смонтировать и получить доступ к старым резервным данным. Даже популярные файловые системы могут поддерживаться не полностью. Например, файловая система reiserfs не полностью поддерживается операционной системой Kubuntu, по крайней мере на момент написания книги.

Создание файловых систем и управление ими

Ubuntu позволяет использовать два варианта разбиения жесткого диска: автоматически, используя программу-установщик и схему по умолчанию, и вручную в процессе первой установки Linux. Программа-установщик позволяет определить, необходимо ли форматировать весь жесткий диск и разделы, предназначенные для установки Linux, или же для создания необходимых разделов достаточно использовать исключительно свободное место на жестком диске. Для выполнения ручной настройки необходимо выбрать параметр, позволяющий осуществить индивидуальную разбивку.

При разбивке диска вручную приложение для разбиения жесткого диска (некогда называемое Disk Druid) предоставляет возможность самостоятельно разбивать жесткий диск на разделы. Позже появилось множество консольных утилит, позволяющих работать с разделами жесткого диска и файловыми системами, создаваемыми на этих разделах.

Разбиение жестких дисков

Исторически жесткие диски использовали 32-битные таблицы разбиения PC-BIOS с головной загрузочной записью (MBR), что ограничивало размер разделов до 2 Тбайт и позволяло создавать лишь четыре основных раздела на одном жестком диске. Использование расширенных разделов сняло последнее ограничение. Для преодоления границы в 2 Тбайт таблицы разбиения PC-BIOS были заменены GPT (таблицами разбиения GUID).

Стандартной командой, предназначенной для работы с разделами жесткого диска, является команда `fdisk`, но, поскольку она не может работать с разделами GPT, она постепенно уступает место другим командам. Более эффективным и поддерживаемым приложением является команда `parted`.

ПРИМЕЧАНИЕ

Если для разбиения, изменения размера и выполнения других операций с жестким диском вы предпочитаете использовать графические приложения, то можете попробовать приложения gparted или qtparted. Названия команд и программных пакетов одинаковы для обоих приложений, но ни одно из них не устанавливается по умолчанию.

Разбиение жесткого диска с помощью команды fdisk

Команда fdisk является весьма полезным инструментом для просмотра и изменения разде́лов дисков. Однако помните, что изменение или удаление разделов может уничтожить ценные данные, хранящиеся на диске, поэтому, прежде чем выполнять какие-либо действия, убедитесь в их необходимости. Чтобы использовать команду fdisk для **просмотра информации о разделах жесткого диска**, выполните следующую команду от имени суперпользователя:

```
$ sudo fdisk -l                   Отображает все разделы каждого жесткого диска
Disk /dev/sda: 82.3 GB, 82348277760 bytes
255 heads, 63 sectors/track, 10011 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
   Device Boot  Start    End   Blocks     Id    System
/dev/sda1   *       1      3   104391     83    Linux
/dev/sda2          14   9881   79264710   83    Linux
/dev/sda3        9882  10011   1044225    82    Linux swap
```

В данном примере выводится информация о жестком диске размером 80 Гбайт, который разбит на три раздела. Первый раздел (/dev/sda1) является небольшим загрузочным разделом /boot, настроенным в файловой системе ext3 (Id 83). Стоит отметить, что знак * обозначает, что первый раздел является загрузочным. Следующий раздел предназначен для корневой файловой системы и также относится к ext3. Последний раздел — раздел подкачки Linux.

ПРИМЕЧАНИЕ

Начиная с версии 2.6.20 ядра Linux, и IDE-, и SCSI-диски используют имена устройств в формате /dev/sd?, где знак ? заменяется буквой (a, b, c и т. д.). В более старых версиях Ubuntu только SCSI-диски и USB-накопители используют имена /dev/sd?. Жесткие диски IDE используют формат имени /dev/hd?.

Если на вашем компьютере установлено несколько дисков, команда fdisk -l отобразит информацию обо всех дисках, если **вы не укажете конкретный**:

```
$ sudo fdisk -l /dev/sdb     Выводит информацию о разделах выбранного диска
```

Чтобы с помощью команды fdisk **работать с конкретным диском**, укажите, с каким диском необходимо работать, не используя никаких параметров:

```
$ sudo fdisk /dev/sda        Выполняет команду fdisk в интерактивном
                             режиме для первого диска
Command (m for help): m      Для отображения помощи нажмите клавишу m
Command action
   a toggle a bootable flag
   b edit bsd disklabel
   c toggle the dos compatibility flag
   d delete a partition
   l list known partition types
   m print this menu
   n add a new partition
   o create a new empty DOS partition table
   p print the partition table
   q quit without saving changes
```

```
   s create a new empty Sun disklabel
   t change a partition's system id
   u change display/entry units
   v verify the partition table
   w write table to disk and exit
   x extra functionality (experts only)
Command (m for help):
```

На основании выведенной на экран информации вы можете использовать любую указанную команду для работы с жестким диском. В частности, вы можете использовать параметр p (выполняющий ту же функцию, что и fdisk -l), n (для создания нового раздела), d (для удаления существующего раздела), l (для отображения известных типов файловых систем) или t (для изменения типа файловой системы раздела). В следующих примерах показаны некоторые операции с командой fdisk:

```
Command (m for help): d                      Запрашивает разрешение
                                             на удаление раздела
Partition number (1-4): 4                    Отображает номер удаляемого
                                             раздела
Command (m for help): n                      Создает новый раздел диска
First cylinder (1-4983, default 1): 1        Выбрать первый цилиндр
                                             (или нажмите клавишу Enter)
Last cylinder ... (default 4983): 4983       Выбрать последний цилиндр
                                             (или нажмите клавишу Enter)
Command (m for help): a                      Делает раздел загрузочным
Partition number (1-3): 1                    Введите номер загрузочного раздела
Command (m for help): t                      Выберите тип файловой системы
Partition number (1-3): 3                    Выберите раздел для изменения
Hex code (type L to list codes): 82          Сделать раздел разделом подкачки
```

Если вы не укажете иного, команда fdisk задаст тип нового раздела как ext3 (83). Для отображения списка типов файловых систем, отображаемого с помощью параметра l, а также соответствующих им и шестнадцатеричных кодов вы можете воспользоваться параметром L. Как говорилось выше, число 82 соответствует разделу подкачки. Среди других типов разделов Linux, которые могут вас заинтересовать, можно назвать Linux-расширенный (85), Linux-LVM (8e), Linux-программный массив (fd) и EFI/GTP (ee).

Для разделов Windows вы можете назначать разделы HPFS/NTFS (7), Windows 95 FAT32 (b), FAT16 (6) или Windows 95 FAT32 LBA (c). Среди других подобных типов файловых систем UNIX можно назвать Minix (be или bf), BSD/OS (e4), FreeBSD (ee), OpenBSD (ef), NeXTSTEP (f0), Darwin UFS (f1) и NetBSD (f4). Вам может пригодиться любой из этих типов файловых систем, если у вас имеются старые резервные копии данных тех файловых систем, которые вы хотите восстановить.

До сих пор нами не производилось никаких изменений в таблице разбиения. Если вы абсолютно уверены в правильности вносимых изменений, то для применения изменений к таблице разбиения разделов введите w. Для отмены изменений (или выхода из fdisk после их применения) введите q.

Копирование таблиц разбиения посредством команды sfdisk

Для **создания резервной копии или дублирования таблицы разбиения жесткого диска** используйте команду sfdisk:

`$ sudo sfdisk -d /dev/sda > sda-table`	*Выполняет резервное копирование таблицы разбиения в файл*
`$ sudo sfdisk /dev/sda < sda-table`	*Восстанавливает таблицу разбиения из файла*
`$ sudo sfdisk -d /dev/sda | sfdisk /dev/sdb`	*Копирует таблицу разбиения с одного диска на другой*

Изменение разделов диска с помощью команды parted

Как и fdisk, команда parted может быть использована для отображения или изменения разделов диска. Однако команда parted предоставляет еще некоторые полезные функции. С помощью следующей команды вы можете **отобразить разделы диска /dev/sda**:

```
$ sudo parted /dev/sda print
Model: ATA FUJITSU MPG3409A (scsi)
Disk /dev/sda: 41.0GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Number   Start    End      Size     Type     File system    Flags
1        32.3kB   206MB    206MB    primary    ext3         boot
2        206MB    39.5GB.  39.3GB   primary    ext3
3        39.5GB   41.0GB   1536MB   primary    linux-swap
```

Из приведенной информации видно, что диск имеет классическую метку (или таблицу разбиения диска) msdos или gpt. В данном случае приведена таблица разбиения msdos.

Чтобы **выполнить команду parted в интерактивном режиме**, введите parted, а затем, если у вас несколько устройств, имя соответствующего устройства хранения информации, с которым хотите работать (например, /dev/sda):

```
$ sudo parted
GNU Parted 1.7.1
Using /dev/sda
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted)
```

При интерактивном использовании команды parted вы можете как целиком вводить команды, так и набирать только несколько первых букв команд и нажимать клавишу **Tab** для их дополнения (в консоли bash). Если же вы хорошо знакомы с используемыми командами, то можете просто вводить необходимое количество букв, чтобы команда parted определяла по ним необходимую для выполнения команду (в Cisco IOS): p для вывода, mkl для mklabel и т. д.

ВНИМАНИЕ

В отличие от fdisk, parted сразу же применяет изменения к разделам, не выполняя предварительную запись на диск, поэтому не рассчитывайте на возможность отмены внесенных изменений, прервав выполнение команды parted.

Для каждой команды `parted` предусмотрена возможность использования со всеми аргументами (например, `mkpart logical ext3 10.7GB 17.0GB`), однако вы можете просто выполнить необходимую команду (`mkpart`), и `parted` переведет вас в интерактивный режим:

```
(parted) mkpart       Создает новый раздел
Partition type? [logical]? primary
File system type? [ext2]? ext3
Start? 17GB
End? 24GB
```

Старайтесь не использовать команду `mkpartfs`, поскольку она не может правильно создавать разделы ext3. Вместо нее для создания раздела ext3 воспользуйтесь командой `mkpart` (как показано выше), а затем отформатируйте его с помощью `parted`, используя команду `mkfs.ext3`. Вообще, поддержка файловых систем типа ext3 у `parted` отсутствует. **Изменение размеров общих разделов Linux** может понадобиться, если нужно освободить место для нового раздела, например:

```
(parted) resize 2           Изменяет размер раздела
Start? [1.2GB] 1.2GB
End? [24GB] 10GB
```

ВНИМАНИЕ

Если вы не используете LVM, эта команда просто разрушит вашу систему.

Для **изменения размера разделов NTFS** предназначена команда `ntfsresize`. В Ubuntu эта команда поставляется с программным пакетом ntfsprogs, в котором также присутствуют команды для создания (`mkfs.ntfs`), настройки (`ntfsfix`) и получения информации о разделах NTFS (`ntfsinfo`).

Работа с метками файловых систем

Термин «*метка*» в отношении разделов дисков может использоваться для обозначения двух разных вещей: *меткой диска* называется таблица разбиения (как видно из возвращаемой командой `parted` информации), а *метка раздела* используется для обозначения имени отдельного раздела. Чтобы **просмотреть метку раздела**, воспользуйтесь командой `e2label`:

```
$ sudo e2label /dev/sda2
/home
```

Чтобы **назначить метку разделу**, выполните следующую команду:

```
$ sudo e2label /dev/sda2 mypartition
```

Не забывайте, что каталог /etc/fstab, как показано ниже, иногда использует метку раздела для монтирования раздела. Изменение этой метки может привести к сбоям при загрузке системы.

```
LABEL=/boot             /boot ext3        defaults    1 2
```

Чтобы **найти раздел, если известна только его метка**, выполните следующую команду:

```
$ sudo findfs LABEL=mypartition
/dev/sda2
```

Форматирование файловой системы

Если вы используете несколько разделов дисков, то можете на каждом из них создать отдельную файловую систему. В большинстве операционных систем Linux для создания и проверки файловых систем используются стандартные для Linux команды. Команды mkfs и fsck позволяют форматировать и проверять файловые системы соответственно.

Команда mkfs выступает в качестве хранилища многих команд, предназначенных для форматирования отдельных типов файловых систем, например mkfs.ext2, mkfs.ext3, mkfs.cramfs, mkfs.msdos, mkfs.ntfs и mkfs.vfat. Если к этим командам добавить пакеты, поддерживающие другие файловые системы, то для беспрепятственной работы с mkfs будут доступны дополнительные команды, среди которых можно назвать mkfs.bfs, mkfs.minix, mkfs.xfs и mkfs.xiafs, допускающие как непосредственное свое выполнение (например, mkfs.vfat /dev/sdb1), так и посредством команды mkfs (например, mkfs -t vfat /dev/sdb1).

Создание файловой системы на разделе жесткого диска

Среди основных программных пакетов, необходимых для создания и проверки файловой системы Ubuntu, можно назвать util-linux (включает в себя команду mkfs и другие приложения общего назначения) и e2fsprogs (включает в себя специальные приложения файловых систем ext2/ext3). Специальные команды mkfs для различных типов файловых систем входят в программные пакеты ntfsprogs (предназначен для работы в NTFS), dosfstools (MS-DOS и VFAT), xfsprogs (XFS), jfsutils (JFS), mtd-utils (JFFS и JFFS2) и reiserfs-utils (reiserfs). Основные приложения устанавливаются вместе с Ubuntu.

Ниже приведены примеры использования команды mkfs, позволяющие **создавать файловые системы** (убедитесь, что добавлен параметр -t):

```
$ sudo mkfs -t ext3 /dev/sdb1          Создает файловую систему ext3
                                       на разделе sba1
$ sudo mkfs -t ext3 -v -c /dev/sdb1    Расширенная команда, выполняющая
                                       также поиск поврежденных блоков
$ sudo mkfs.ext3 -c /dev/sdb1          То же
```

Параметр -L позволяет добавить метку новому разделу:

```
$ sudo mkfs.ext3 -c -L mypartition /dev/sdb1   Добавляет разделу метку
```

Создание виртуальной файловой системы

Если вы хотите лишь ознакомиться с различными типами файловых систем или сделать файловую систему более гибкой (другими словами, не привязанной к физическому диску), то можете создать *виртуальную файловую систему*. Виртуальная файловая система — это файловая система, не создающая файлы на существующей файловой системе, однако вы можете форматировать ее так же, как и файловую систему любого другого типа, перемещаться по ней и получать к ней доступ с других компьютеров.

Виртуальные файловые системы могут использоваться, например, при создании Live CD или запуске выделенных виртуальных файловых систем. В следующем примере создается пустой файл-образ диска объемом 500 Мбайт, затем он форматируется как файловая система и монтируется, чтобы обеспечить доступ к данным файловой системы:

```
$ dd if=/dev/zero of=mydisk count=2048000     Создает заполненный нулями
                                              файл обьемом 1 Гбайт
$ du -sh mydisk                               Проверяет размер виртуальной
                                              файловой системы
1001M mydisk
$ mkfs -t ext3 mydisk                         Создает файловую систему
                                              на mydisk
mydisk is not a block special device
Continue (y/n): y
$ sudo mkdir /mnt/image                       Создает точку монтирования
$ sudo mount -o loop mydisk /mnt/image        Монтирует файл mydisk
                                              в каталог /mnt/image
```

В данном примере команда dd создает пустой файл-образ диска размером 2 048 000 блоков (приблизительно 1 Гбайт). Затем команда mkfs создает файловую систему ext3 (вообще же команда mkfs может создать файловую систему любого необходимого типа). Поскольку файл не является специальным блочным устройством, как в предыдущем случае форматирования разделов диска, mkfs предупредит вас о начале создания новой файловой системы. После создания точки монтирования команда сообщит, что вы смонтировали файл (mydisk) как петлевое устройство (-o loop). Стоит отметить, что из вышеуказанных команда mount является единственной, требующей привилегий суперпользователя.

После завершения монтирования виртуальной файловой системы вы сможете получать к ней доступ из любой файловой системы. Когда вы закончите работать с файловой системой, выйдите из нее и размонтируйте:

```
$ sudo cd /mnt/image            Переходит к точке монтирования
$ sudo mkdir test               Создает каталог в файловой системе
$ sudo cp /etc/hosts            Копирует файл в файловую систему
$ cd                            Выходит из файловой системы
$ sudo umount /mnt/image        Демонтирует файловую систему
```

После завершения процесса демонтирования виртуальной файловой системы вы можете переместить ее в другую файловую систему или записать на компакт-диск

для последующего использования в другом месте. Если вам больше не нужна файловая система, просто удалите соответствующий файл.

Просмотр и изменение атрибутов файловой системы

С помощью команд tune2fs и dumpe2fs вы можете **просматривать атрибуты файловых систем ext2 и ext3**. Команда tune2fs также может быть использована для изменения атрибутов файловой системы. Для создания же раздела подкачки используйте команду swapfs:

```
$ sudo tune2fs -l /dev/sda1        Отображает настраиваемые атрибуты файловой
                                   системы
$ sudo dumpe2fs -h /dev/sda1       То же
dumpe2fs 1.39 (29-May-2006)
Filesystem volume name: /
Last mounted on: <not available>
Filesystem UUID: f5f261d3-3879-41d6-8245-f2153b003204
Filesystem magic number: 0xEF53
Filesystem revision #: 1 (dynamic)
Filesystem features: has_journal ext_attr resize_inode dir_index filetype
   needs_recovery sparse_super large_file
Default mount options: user_xattr acl
Filesystem state: clean
Errors behavior: Continue
Filesystem OS type: Linux
Inode count: 7914368
Block count: 7907988
Reserved block count: 395399
Free blocks: 5916863
Free inodes: 7752077
First block: 0
Block size: 4096
Fragment size: 4096
Reserved GDT blocks: 1022
Blocks per group: 32768
Fragments per group: 32768
Inodes per group: 32704
Inode blocks per group: 1022
Filesystem created: Fri Jun 15 12:13:17 2007
Last mount time: Tue Jul 24 06:47:35 2007
Last write time: Tue Jul 24 06:47:35 2007
Mount count: 2
Maximum mount count: 29
Last checked: Fri Jun 15 12:13:17 2007
Check interval: 0 (<none>)
Reserved blocks uid: 0 (user root)
Reserved blocks gid: 0 (group root)
First inode: 11
```

```
Inode size: 128
Journal inode: 8
First orphan inode: 988413
Default directory hash: tea
Directory Hash Seed: 4137d20d-b398-467b-a47a-a9110416b393
Journal backup: inode blocks
Journal size: 128M
```

Данные команды отображают большой объем информации о файловой системе. Например, если вы используете файловую систему, для которой нужно создавать большое количество небольших файлов (например, новостной сервер), то с помощью этих команд можете проверять, остались ли у вас файлы inode. Значение поля Maximum mount count определяет количество проверок файловой системы после ее монтирования. Среди прочей информации вы можете также найти дату и время создания файловой системы, ее последнего монтирования, а также изменения.

Чтобы **изменить настройки существующей файловой системы ext2 или ext3**, воспользуйтесь командой tune2fs. Следующая команда изменяет количество необходимых монтирований файловой системы перед ее принудительной проверкой:

```
$ sudo tune2fs -c 31 /dev/sda1    Устанавливает количество монтирований равным #
tune2fs 1.39 (29-May-2006)
Setting maximal mount count to 31
```

Если же вы хотите выполнять проверки файловой системы **на основе заданных временных интервалов**, а не количестве монтирований, отключите проверку mount-count, установив его в значение -1:

```
$ sudo tune2fs -c -1 /dev/sda1
tune2fs 1.39 (29-May-2006)
Setting maximal mount count to -1
```

Для активизации **динамической (с временно́й зависимостью) проверки** используйте параметр -i:

```
$ sudo tune2fs -i 10 /dev/sda1         Проверяет систему каждые 10 дней
$ sudo tune2fs -i 1d /dev/sda1         Проверяет систему ежедневно
$ sudo tune2fs -i 3w /dev/sda1         Проверяет систему каждые 3 недели
$ sudo tune2fs -i 6m /dev/sda1         Проверяет систему каждые 6 месяцев
$ sudo tune2fs -i 0 /dev/sda1          Отключает динамическую проверку
```

У вас всегда должна быть включена либо mount-count, либо динамическая проверка.

Чтобы **преобразовать файловую систему ext2 в ext3**, используйте параметр -j (с записью действия в журнал):

```
$ sudo tune2fs -j /dev/sda1     Включает журналирование изменения ext2 на ext3
```

Создание и использование разделов подкачки

Для хранения избыточных данных системной оперативной памяти в операционных системах Linux используются разделы подкачки. Если вы не создали раздел

подкачки при установке Linux, то можете создать его позже, воспользовавшись командой mkswap. **Раздел подкачки может быть создан** либо на стандартном разделе жесткого диска, либо на специально отформатированном для этого разделе:

```
$ sudo mkswap /dev/sda1          Форматирует раздел sda1 как раздел подкачки
Setting up swapspace version 1, size = 205594 kB
```

Чтобы **проверить раздел подкачки на наличие поврежденных блоков**, вместе с командой mkswap используйте параметр -c:

```
$ sudo mkswap -c /dev/sda1
```

Если у вас нет свободного раздела, то в **качестве области подкачки можете использовать файл**:

```
$ sudo dd if=/dev/zero of=/tmp/swapfile count=65536
65536+0 records in
65536+0 records out
33554432 bytes (34 MB) copied, 1.56578 s, 21.4 MB/s
$ sudo chmod 600 /tmp/swapfile
$ sudo mkswap /tmp/swapfile
Setting up swapspace version 1, size = 67104 kB
```

В данном примере команда dd создает файл swapfile размером 32 Мбайт. Затем, чтобы во время работы не отображались предупреждения от команды swapon, команда chmod закрывает к этому файлу доступ. После этого, чтобы назначить файл /tmp/swapfile разделом подкачки, команда mkswap форматирует его.

После создания раздела или файла подкачки необходимо, воспользовавшись командой swapon, **заставить систему использовать созданную область подкачки**:

```
$ sudo swapon /dev/sda1       Задействует раздел /dev/sda1 в качестве
                              раздела подкачки
$ sudo swapon -v /dev/sda1    Задействует раздел в качестве раздела
                              подкачки и отображает больше информации о нем
swapon on /dev/sda1
$ sudo swapon -v /tmp/swapfile    Задействует файл /tmp/swapfile в качестве
                                  раздела подкачки
swapon on /tmp/swapfile
```

Кроме того, вы можете использовать команду swapon и **для просмотра списка файлов и разделов подкачки**:

```
$ swapon -s              Отображает список всех задействованных
                         файлов и разделов подкачки
Filename        Type        Size                Used      Priority
/dev/sda5       partition   1020088             142764    -1
/tmp/swapfile   file        65528       0                 -6
```

Чтобы **деактивировать область подкачки**, воспользуйтесь командой swapoff:

```
$ sudo swapoff -v /tmp/swapfile
swapoff on /tmp/swapfile
```

Области подкачки используются в соответствии с их приоритетом. Ядро в первую очередь задействует области с высоким приоритетом, а затем использует

остальные. Области с одинаковым приоритетом задействуются одновременно и могут пересекаться. С помощью параметра -p вы можете определить приоритет областей подкачки:

```
$ sudo swapon -v -p 1 /dev/sda1    Назначает разделу sda1 высший
                                   приоритет подкачки
```

Монтирование и демонтирование файловых систем

Перед использованием обычной, не предназначенной для подкачки файловой системы необходимо добавить ее в каталог в дереве файловых систем вашего компьютера, смонтировав ее. Ваша корневая файловая система (/), а также другие файловые системы, которые вы постоянно используете, обычно монтируются автоматически на основе записей в файле /etc/fstab. Другие файловые системы могут быть смонтированы вручную по мере надобности с помощью команды mount.

Монтирование файловых систем из файла fstab

Во время первой установки Linux файл /etc/fstab, содержащий информацию о вашей корневой файловой системе и других файловых системах, обычно устанавливается автоматически. После этого данные файловые системы могут монтироваться автоматически во время загрузки системы либо вручную при необходимости (с готовыми к использованию точками монтирования и другими параметрами).

Рассмотрим пример файла /etc/fstab:

```
/dev/VolGroup00/LogVol00   /            ext3     defaults        1 1
LABEL=/boot                /boot        ext3     defaults        1 2
tmpfs                      /dev/shm     tmpfs    defaults        0 0
devpts                     /dev/pts     devpts   gid=5,mode=62   0 0 0
sysfs                      /sys         sysfs    defaults        0 0
proc                       /proc        proc     defaults        0 0
/dev/VolGroup00/LogVol01   swap         swap     defaults        0 0
/dev/sda1                  mnt/windows  vfat     noauto          0 0
```

ПРИМЕЧАНИЕ

Чтобы вам было понятнее, в приведенном выше примере для каждой файловой системы, имеющейся на жестком диске, список номеров UUID был удален. Для любой файловой системы эти номера имеют вид UUID=da2dbc48-862e-4fbe-9529-a88b57b15bac и записываются перед типом файловой системы.

Как видно из приведенного примера, все файловые системы, за исключением /dev/sda1 (параметр noauto), монтируются автоматически. Корневой раздел (/) и раздел подкачки (swap) настраиваются как тома LVM (управление логическими томами), которые облегчают удаление или добавление физических разделов, содержащих ID томов. Среди файловых псевдосистем (не связанных с каким-либо разделом) можно назвать devpts (интерфейс для псевдотерминалов pty), sysfs

(информация ядра 2.6) и proc (информация более ранних версий ядра). Раздел диска /dev/sda1 в данном примере был добавлен вручную и предназначен для монтирования раздела Windows, хранящегося на данном устройстве.

Файл /etc/fstab здесь не является хранилищем информации о съемных носителях, используемым по умолчанию, поэтому Hardware Abstraction Layer (HAL) автоматически определяет съемные носители и монтирует их в специально отведенные точки монтирования каталога /media (на основании ID тома на носителе).

В табл. 7.2 приведено описание полей файла /etc/fstab.

Таблица 7.2. Поля файла /etc/fstab

Поле	Описание
1	Имя устройства, содержащего файловую систему. По умолчанию данное поле содержит имя устройства раздела для монтирования (например, /dev/sda1), однако значение этого поля также может быть равно LABEL или содержать уникальный идентификатор (UUID)
2	Указывает точку монтирования. Файловая система содержит все данные, начиная от информации о точке монтирования и заканчивая информацией о структуре дерева каталогов, если в данной точке не была ранее смонтирована другая файловая система
3	Указывает тип файловой системы. Для получения большей информации об основных типах файловых систем ознакомьтесь с табл. 7.1
4	Содержит параметры команды mount, среди которых можно отметить noauto (используется для предотвращения монтирования файловой системы во время загрузки) и ro (монтирует файловую систему, предназначенную только для чтения). Чтобы дать возможность всем пользователям монтировать файловую систему, добавьте в это поле параметр user (пользователь) или owner (владелец). Параметры должны быть разделены запятыми. Для получения более подробной информации о поддерживаемых параметрах обратитесь к странице справочника, посвященной команде mount
5	Данное поле содержит значение только в том случае, если вы использовали для выполнения резервного копирования данных команду dump. Значение 1 означает, что файловая система должна быть разгружена, 0 — что разгрузку выполнять не следует
6	Число в этом поле определяет, должна ли файловая система быть проверена с помощью команды fsck. Значение 0 означает, что файловая система не нуждается в проверке, значение 1 — что файловую систему необходимо проверить в первую очередь (используется в качестве корневой файловой системы), значение 2 — что файловая система может быть проверена после завершения проверки корневой файловой системы

Вы можете создавать собственные записи в файле /etc/fstab для разделов любого жестко диска или съемного носителя. Удаленные файловые системы (NFS, Samba и др.) также могут содержать записи в файле /etc/fstab, чтобы автоматически монтироваться во время загрузки или позже вручную.

Команда mount

Команда mount предназначена для просмотра смонтированных файловых систем, а также для монтирования любых локальных (жесткий диск, запоминающее устройство USB, компакт-диск, DVD и т. д.) или удаленных (NFS, Samba и т. д.)

файловых систем. Ниже приведены примеры использования команды mount для **просмотра смонтированных файловых систем**:

```
$ mount          Отображает список смонтированных, удаленных
                 и локальных файловых систем
/dev/sda7 on / type ext3 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
/dev/sda6 on /mnt/debian type ext3 (rw)
/dev/sda3 on /mnt/slackware type ext3 (rw)
tmpfs on /dev/shm type tmpfs (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
```

Для **отображения только конкретных смонтированных файловых систем** используйте параметр -t:

```
$ mount -t ext3        Отображает смонтированные ext3-файловые системы
/dev/sda7 on / type ext3 (rw)
/dev/sda6 on /mnt/debian type ext3 (rw)
/dev/sda3 on /mnt/slackware type ext3 (rw)
```

Для **отображения меток разделов с информацией о монтировании** используйте параметр -l:

```
$ mount -t ext3 -l          Отображает смонтированные ext3-файловые
                            системы и их метки
/dev/sda7 on / type ext3 (rw) [/123]
/dev/sda6 on /mnt/debian type ext3 (rw) [/mnt/debian]
/dev/sda3 on /mnt/slackware type ext3 (rw) [/mnt/slackware]
```

Ниже представлен пример использования команды mount для монтирования устройства /dev/sda1 в существующий каталог /mnt/mymount:

```
$ sudo mount /dev/sda1 /mnt/mymount/     Монтирует локальную файловую систему
$ sudo mount -v /dev/sda1 /mnt/mymount/  Монтирут файловую систему, выводит
                                         более подробную информацию
mount: you didn't specify a filesystem type for /dev/sda1
I will try type ext3
/dev/sda1 on /mnt/mymount type ext3 (rw)
```

В данном примере команда mount ищет записи в файле/etc/fstab каталога /dev/sda1 либо же пытается определить тип смонтированной файловой системы.

Для **точного определения типа файловой системы, предназначенной для монтирования**, используйте параметр -t:

```
$ sudo mount -v -t ext3 /dev/sda1 /mnt/mymount/     Монтирует файловую
                                                    систему ext3
/dev/sda1 on /mnt/mymount type ext3 (rw)
```

Вы также можете **отобразить метку и имя монтируемого раздела**:

```
$ sudo mount -vl -t ext3 /dev/sda1 /mnt/mymount/    Монтирует файловую систему
                                                    и отображает ее метку
```

Если вы захотите смонтировать раздел, информация о котором уже содержится в файле fstab, то используйте либо точку монтирования, либо имя соответствующего устройства. Например, с помощью записи

```
/dev/sda1 /mnt/mymount      ext3        defaults    1 2
```

в файле fstab вы можете смонтировать файловую систему любым из предложенных ниже способов:

```
$ sudo mount -v /dev/sda1          Монтирует файловую систему по имени
                                   устройства хранения
/dev/sda1 on /mnt/mymount type ext3 (rw)
$ sudo mount -v /mnt/mymount/      Монтирует файловую систему на основании
                                   точки монтирования
/dev/sda1 on /mnt/mymount type ext3 (rw)
```

Добавив параметр -o и разделив запятыми список параметров, вы можете **определить параметры монтирования раздела**. К этим параметрам относятся те же, которые могут быть добавлены в четвертое поле файла /etc/fstab. По умолчанию разделы монтируются с правами чтения и записи, однако вы можете точно определить, **монтировать ли файловую систему для чтения и записи (rw) или только для чтения (ro)**:

```
$ sudo mount -v -t ext3 -o rw /dev/sda1 /mnt/mymount/ Монтирует раздел
                                                      с правами чтения и записи
/dev/sda1 on /mnt/mymount type ext3 (rw)
$ sudo mount -v -t ext3 -o ro /dev/sda1 /mnt/mymount/ Монтирует раздел
                                                      с правами только чтения
/dev/sda1 on /mnt/mymount type ext3 (ro)
```

Команда mount может использовать еще несколько полезных параметров.

- noatime — не обновляет время доступа к файлам, полезен при работе в файловых системах с большим количеством входящих и исходящих потоков, таких как почтовые серверы и журналы.
- noexec — предотвращает выполнение бинарных файлов, хранящихся на данной файловой системе. Параметр может использоваться для повышения безопасности, например для каталога /tmp в среде с неавторизованными пользователями.
- remount — изменяет параметры монтированной файловой системы. С помощью этого параметра вы можете размонтировать файловую систему, а потом снова ее смонтировать с уже новыми параметрами. В следующем примере изменяются права чтения и записи только на чтение:

```
$ sudo mount -v -o remount,ro /dev/sda1
/dev/sda1 on /mnt/mymount type ext3 (ro)
```

- --bind — монтирует существующую файловую систему, используя другую точку монтирования. Если раздел /dev/sda1 уже смонтирован в каталог /mnt/mymount, выполните следующее:

```
$ sudo mount --bind -v /mnt/mymount/ /tmp/mydir/
/mnt/mymount on /tmp/mydir type none (rw,bind)
```

Теперь одна и та же файловая система доступна по двум адресам. Новая точка монтирования обладает теми же параметрами, что и оригинальная.

- --move — перемещает файловую систему из одной точки монтирования в другую. Если раздел /dev/sda1 уже смонтирован в каталог /mnt/mymount, то использование данного параметра переместит файловую систему в каталог /tmp/mydir:

```
$ sudo mount -v --move /mnt/mymount/ /tmp/mydir/
/mnt/mymount on /tmp/mydir type none (rw)
```

Аналогично файлу подкачки вы можете создать файловую систему в файле, а затем смонтировать ее. Данная процедура называется *петлевым* монтированием. Порядок создания и монтирования файла описан в подразд. «Создание виртуальной файловой системы» разд. «Создание файловых систем и управление ими». Обычно **петлевое монтирование файла используется** после загрузки установочного компакт-диска или Live CD Linux. Выполнив петлевое монтирование образа компакт-диска, вы можете просматривать его содержимое и копировать файлы с него на свой жесткий диск.

В следующем примере команда mount автоматически выбирает существующее петлевое устройство при монтировании файла-образа компакт-диска (тип файловой системы iso9660). Из возвращаемой командой информации видно, что было выбрано устройство /dev/loop0:

```
$ sudo mount -v -t iso9660 -o loop /tmp/myimage.iso /mnt/mymount/
mount: going to use the loop device /dev/loop0
/tmp/myimage.iso on /mnt/mymount type ext3 (rw,loop=/dev/loop0)
```

В следующем примере загрузочный образ запоминающего устройства USB Linux diskboot.img загружается в каталог /tmp. Ниже приведен пример **монтирования загрузочного образа**:

```
$ sudo mount -v -o loop /tmp/diskboot.img /mnt/mymount
mount: going to use the loop device /dev/loop0
mount: you didn't specify a filesystem type for /dev/loop0
I will try type vfat
/tmp/diskboot.img on /mnt/mymount type vfat (rw,loop=/dev/loop0)
```

Для **просмотра состояния петлевых устройств** используйте команду losetup:

```
$ sudo losetup /dev/loop0          Отображает список смонтированных
                                   петлевых устройств
/dev/loop0: [0807]:1009045 (/tmp/diskboot.img)
```

Если петлевое монтирование прерывается или у вас возникают проблемы в ходе демонтирования раздела, попробуйте выполнить следующую команду:

```
$ sudo losetup -d /dev/loop1       Принудительное демонтирование
                                   смонтированного петлевого устройства
```

РИМЕЧАНИЕ

Команда mount может также использоваться для привязки к ресурсам NFS или Samba/Windows CIFS. Для получения более подробной информации о монтировании этих типов удаленных файловых систем обратитесь к гл. 12.

Демонтирование файловых систем

Для размонтирования файловых систем предназначена команда umount. Вы можете осуществлять демонтирование файловых систем, используя имя устройства или точку монтирования, однако точку монтирования использовать предпочтительнее, поскольку это позволяет избежать проблем при связанном монтировании (одно устройство, несколько точек монтирования):

```
$ sudo umount -v /dev/sda1          Размонтирует устройство по его имени
/dev/sda1 umounted
$ sudo umount -v /mnt/mymount/      Размонтирует устройство по точке
                                    монтирования
/tmp/diskboot.img umounted
```

Если устройство занято, размонтировать его не удастся. Наиболее распространенной причиной, препятствующей размонтированию устройства, является открытая консоль с текущим каталогом в зоне монтирования:

```
$ sudo umount -v /mnt/mymount/
umount: /mnt/mymount: device is busy
umount: /mnt/mymount: device is busy
```

Иногда приложение, использующее устройство, неизвестно. В этом случае для просмотра списка открытых файлов и последующего **поиска в этом списке интересующей точки монтирования** вы можете воспользоваться командой lsof:

```
$ sudo lsof | grep mymount          Ищет открытые файлы в разделе mymount
bash     9341  francois    cwd    DIR   8,1   1024  2  /mnt/mymount
```

Вы можете видеть, что процесс bash, запущенный пользователем francois с PID 9341, препятствует размонтированию раздела mymount.

Другой причиной может являтся выполнение побочного размонтирования:

```
$ sudo umount -vl /mnt/mymount/     Выполнение побочного размонтирования
```

Побочное размонтирование в данный момент размонтирует файловую систему из дерева, но перед началом очистки ожидает полного освобождения устройства. Размонтирование съемных носителей также можно осуществлять с помощью команды eject, которая **размонтирует компакт-диск и извлекает его из привода**:

```
$ sudo eject /dev/cdrom             Размонтирует и извлекает компакт-диск
```

Проверка файловых систем

В Linux вместо обычного приложения scandisk, используемого в Windows, поиск поврежденных на физическом уровне блоков осуществляется с помощью команды badblocks, а поиск ошибок на логическом уровне — с помощью команды fsck. Ниже описывается способ **осуществления поиска поврежденных блоков**:

```
$ sudo badblocks /dev/sda1          Осуществляет поиск физически
                                    поврежденных блоков на жестком диске
```

```
$ sudo badblocks -v /dev/sda1     Отображает подробную информацию
                                   о сканировании жесткого диска
Checking blocks 0 to 200781
Checking for bad blocks (read-only test): done
Pass completed, 0 bad blocks found.
```

По умолчанию команда badblock тестирует поврежденные блоки на чтение. Кроме того, с помощью этой команды вы можете выполнять безопасное тестирование секторов на чтение и запись. Этот тест наиболее длителен, однако позволяет сохранять хранящиеся на устройстве данные. Для **контроля выполнения команды** добавьте параметр -s:

```
$ sudo badblocks -vsn /dev/sda1     Выполняет безопасную проверку дисков
                                     на наличие поврежденных блоков
Checking for bad blocks in non-destructive read-write mode
From block 0 to 200781
Testing with random pattern: Pass completed, 0 bad blocks found.
```

Следующая команда выполняет более быстрый, однако уничтожающий данные тест чтения-записи:

ВНИМАНИЕ

Эта команда уничтожит все данные на проверяемом разделе.

```
$ sudo badblocks -vsw /dev/sda1   Проверяет диск на наличие поврежденных
                                   секторов, уничтожая записанные данные
Checking for bad blocks in read-write mode
From block 0 to 200781
Testing with pattern 0xaa: done
Reading and comparing: done
Testing with pattern 0x55: done
Reading and comparing: done
Testing with pattern 0xff: done
Reading and comparing: done
Testing with pattern 0x00: done
Reading and comparing: done
Pass completed, 0 bad blocks found.
```

Вы можете выполнять сразу **несколько тестов на наличие поврежденных секторов**; например, следующая команда может быть использована для записи компакт-дисков и одновременного вывода на экран информации о поврежденных секторах:

```
$ sudo badblocks -vswp 2 /dev/sda1
```

Как и mkfs, команда fsck представляет собой лишь устройство сбора данных отдельных утилит файловых систем. Чтобы **проверить файловую систему ext3** с помощью команды fsck, просто добавьте к ней имя устройства того раздела диска, который вы хотите проверить:

```
$ sudo fsck /dev/sda1
fsck 1.39 (29-May-2006)
```

```
e2fsck 1.39 (29-May-2006)
mypart has gone 18 days without being checked, check forced.
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
mypart: 11/50200 files (9.1% non-contiguous), 12002/200780 blocks
```

Вы можете использовать совместно с командой fsck и **другие параметры**, например -T (скрывает номер версии fsck) и -V (отображает более подробную информацию о выполнении fsck в реальном времени):

```
$ sudo fsck -TV /dev/sda1          Проверяет файловую систему (отображает
                                   подробную информацию, но скрывает версию)
[/sbin/fsck.ext3 (1) -- /dev/sda1] fsck.ext3 /dev/sda1
e2fsck 1.39 (29-May-2006)
mypart: clean, 11/50200 files, 12002/200780 blocks
```

Если при выполнении команды fsck произойдет какая-либо ошибка, вам будет задан вопрос, хотите ли вы ее исправить:

```
$ sudo fsck -TV /dev/sda1          Отображает вопрос, нужно ли исправлять
                                   возникшие ошибки
[/sbin/fsck.ext3 (1) -- /mnt/mymount] fsck.ext3 /dev/sda1
e2fsck 1.39 (29-May-2006)
Couldn't find ext2 superblock, trying backup blocks...
Resize inode not valid. Recreate<y>? y
```

Если у вас нет достаточного опыта в работе с файловыми системами, отвечайте всегда «да». Добавив к команде параметра -y, этот процесс можно автоматизировать:

```
$ sudo fsck -TVy /dev/sda1
[/sbin/fsck.ext3 (1) -- /mnt/mymount] fsck.ext3 -y /dev/sda1
e2fsck 1.39 (29-May-2006)
Couldn't find ext2 superblock, trying backup blocks...
Resize inode not valid. Recreate? yes
mypart was not cleanly unmounted, check forced.
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
Free blocks count wrong for group #0 (3552, counted=3553).
Fix? yes
Free blocks count wrong (188777, counted=188778).
Fix? yes
mypart: ***** FILE SYSTEM WAS MODIFIED *****
mypart: 11/50200 files (0.0% non-contiguous), 12002/200780 blocks
```

роверка дисков RAID

Технология резервных массивов данных на независимых дисках (RAID) позволяет дублировать и хранить данные на нескольких жестких дисках. Использование технологии RAID может повысить надежность вашего носителя информации. Команда mdadm, являющаяся составной частью пакета mdadm, может использоваться для **проверки устройств softraid**:

```
$ sudo mdadm -Q /dev/md1
/dev/md1: 1498.13MiB raid1 2 devices, 0 spares.
   Use mdadm --detail for more detail.
/dev/md1: No md super block found, not an md component.
```

Сообщение в последней строке свидетельствует о том, что каталог /dev/md1 не является частью массива RAID. Однако это нормально, поскольку сам каталог md1 является массивом данных. Аналогично, если вы с помощью команды mdadm выполните запрос к члену массива RAID, то получите примерно следующую информацию:

```
$ sudo mdadm -Q /dev/sdb3
/dev/sdb3: is not an md array
/dev/sdb3: device 1 in 4 device active raid6 md0. Use mdadm --examine for more detail.
```

Для отображения более подробной информации добавьте параметр --detail:

```
$ sudo mdadm -Q --detail /dev/md1
/dev/md1:
        Version : 00.90.01
  Creation Time : Fri Dec 8 16:32:12 2006
     Raid Level : raid1
     Array Size : 1534080 (1498.38 MiB 1570.90 MB)
    Device Size : 1534080 (1498.38 MiB 1570.90 MB)
   Raid Devices : 2
  Total Devices : 2
Preferred Minor : 1
    Persistence : Superblock is persistent
    Update Time : Sun Jun 17 02:06:01 2007
          State : clean
 Active Devices : 2
Working Devices : 2
 Failed Devices : 0
  Spare Devices : 0
           UUID : 49c564cc:2d3c9a14:d93ce1c9:070663ca
         Events : 0.42
   Number     Major     Minor     RaidDevice   State
   0          3         2         0            active sync  /dev/hda2
   1          3         66        1            active sync  /dev/hdb2
```

Кроме того, команду mdadm можно использовать для управления устройствами softraid. Для получения более подробной информации выполните следующую команду:

```
$ sudo mdadm --manage -help
$ man mdadm
```

ПРИМЕЧАНИЕ

Если вы используете аппаратные RAID-контроллеры 3ware/AMCC, являющиеся, по нашему мнению, наилучшими для приводов SATA RAID, убедитесь, что у вас установлен 3ware Disk Manager (3dm2), доступный в формате RPM на сайте ATrpms.net. Пакет 3dm2 включает в себя инструменты для наблюдения и сетевые GUI.

Перед установкой пакета mdadm убедитесь, что у вас установлен именно RAID-диск. Во время установки mdadm программа попытается произвести настройку ваших RAID-дисков. Кроме того, программа создаст в каталоге /etc/mdadm файл mdadm.conf, с помощью которого вы сможете при необходимости удалить пакет mdadm, однако существование этого файла может стать причиной проблем совместимости с программным пакетом lvm2 package, что будет описано позже.

Получение информации об использовании файловой системы

Недостаточный объем свободного дискового пространства может сказываться на производительности вашей настольной системы, а для ваших серверов это может стать потенциальной катастрофой. Для определения объема доступного в данный момент дискового пространства и уровня его использования предназначена команда df, а для определения занимаемого отдельными файлами и каталогами места — команда du.

Команда df выводит **отчет об использовании смонтированных файловых систем**. С помощью параметра -h для упрощения восприятия выводимых данных вы можете выполнять их пересчет (по умолчанию отображаемых в байтах) в мегабайты (M) и гигабайты (G):

```
$ df -h              Отображает информацию о дисковом пространстве
                     в удобной для восприятия форме
Filesystem           Size       Used       Avail     Use%  Mounted on
/dev/sda2            7.6G       3.4G       3.9G      47%   /
/dev/sda1            99M        14M        80M       15%   /boot
Tmpfs                501M       0          501M      0%    /dev/shm
/dev/sda5            352G       197G       137G      59%   /home
//thompson/chris     9204796    5722608    3007068   66%   /mnt/mymount
```

Поскольку в файловых системах ext существует множество файлов inode, созданных во время выполнения команды mkfs, вы можете перед вычислением объема занимаемого остальными файлами пространства вычислить только объем файлов inode, при условии, что у вас на диске хранится много небольших файлов. Чтобы **проверить объем файлов inode**, воспользуйтесь параметром -i:

```
$ df -hi
Filesystem           Inodes IUsed  IFree       IUse%    Mounted on
/dev/sda2 2.0M       108K          1.9M        6%       /
```

Если вы используете файловые системы, смонтированные для сетевого использования (например, Samba или NFS), они также будут указываться в возвращаемой командой df информации. Чтобы **ограничить объем отображаемой** командой df **информации** данными о локальных файловых системах, выполните следующую команду:

```
$ df -hl        Отображает дисковое пространство, занимаемое
                только локальными файловыми системами
```

Чтобы **добавить в список тип файловой системы**, используйте параметр -T:

```
$ df -hT        Добавляет в список информацию о типе файловой системы
Filesystem Type Size Used Avail Use% Mounted on
/dev/sda7 ext3 8.8G 5.5G 2.9G 66% /
```

Чтобы **определить объем дискового пространства, занимаемого отдельными файлами или каталогами**, используйте команду du. Следующая команда была выполнена от имени пользователя *francois*:

```
$ du -h /home/  Отображает объем занимаемого каталогом /home пространства
du: '/home/chris': Permission denied
4.0K /home/francois/Mail
52K  /home/francois
64K  /home/
```

Из отображенной информации видно, что доступ к одному из каталогов (в данном случае /home/chris) был закрыт из соображений безопасности. В следующих примерах показано, как с помощью учетной записи суперпользователя **обойти систему прав пользователей и получить верные результаты**. Наиболее хорошо это видно, если для отображения отчета использовать параметр -s:

```
$ du -sh /home          Обычному пользователю закрыт доступ к домашним
                        каталогам других пользователей
du: '/home/chris': Permission denied
du: '/home/horatio199': Permission denied
64K  /home
# du -sh /home          С помощью учетной записи суперпользователя вы можете
                        вывести отчет об использовании места на жестком диске
1.6G      /home
```

С помощью параметра -c вы можете **указать только несколько конкретных каталогов**, а затем вывести общий результат:

```
$ sudo du -sch /home /var   Отображает отчет для одного и всех каталогов
1.6G      /home
111M      /var
1.7G      total
```

С помощью параметра exclude вы можете **исключить из поиска файлы**, отождествляемые с образом. В следующем примере файлы-образы дисков (имеющие

расширение ISO) не учитываются в результатах определения свободного места на диске:

```
$ sudo du -sh --exclude='*.iso' /home/chris    Исключает файлы образов ISO
588M      /home/chris
```

Кроме того, вы можете **определить количество проверяемых уровней дерева каталогов**. Чтобы в результатах теста учитывать вложенные каталоги, установите параметр `--max-depth` в значение больше 1:

```
$ sudo du -h --max-depth=1 /home    Определяет использование дискового
                                    пространства до первого уровня
                                    вложенных каталогов
1.6G /home/chris
52K          /home/francois
1.6G /home
$ sudo du -h --max-depth=2 /home    Определяет использование дискового
                                    пространства до второго уровня
                                    вложенных каталогов
...
4.0K      /home/francois/Mail
52K       /home/francois
1.6G      /home
```

Программа управления логическими томами (LVM)

Программа управления логическими томами является приложением операционных систем Linux, предназначенным для того, чтобы помочь вам справляться с изменяющимися потребностями, связанными с управлением дисковым пространством. Настроив свои жесткие диски как тома LVM, вы можете получить невероятную гибкость в управлении дисковым пространством по мере изменения своих потребностей. Кроме того, LVM позволяет фиксировать мгновенное состояние системы (этот параметр обычно можно встретить в SAN (сеть устройств хранения данных) предприятий).

Ubuntu поставляется с LVM2, с помощью которой вы можете получать информацию о размещении разделов жесткого диска уже при первой установке Ubuntu. Используя LVM2, вы можете определять группы томов (vg), логических томов (lv) и физических томов (pv), а также управлять ими. Каждый логический и физический том разделяется на логические и физические составляющие соответственно.

Основное при использовании LVM — создание необходимых групп томов и логических томов, а затем назначение экстентов (небольших фрагментов дискового пространства) в тех участках, где они нужны. В отличие от старых схем разбиения дисков, в которых вам приходилось создавать резервные копии данных, изменять параметры разделов, а затем возвращать данные на измененные разделы, при ис-

пользовании LVM вы можете просто добавлять в необходимые места неиспользованные экстенты.

Чтобы использовать LVM, вы должны установить программный пакет lvm2. LVM поставляется с набором команд, которые могут быть использованы для работы с томами LVM. Изучив материал следующего подраздела, вы сможете больше узнать о командах LVM.

ВНИМАНИЕ

Во избежание порчи жестких дисков компьютера во время изучения LVM мы рекомендуем вам выполнять приводимые далее примеры на второстепенных съемных запоминающих устройствах. Например, для выполнения данных команд мы использовали недорогое запоминающее USB-устройство объемом 32 Мбайт (в /dev/sdb).

Создание томов LVM

Для начала, чтобы создать физические разделы на том запоминающем устройстве, на котором вы хотите создать позже логические разделы, воспользуйтесь командой fdisk. В данном случае используется запоминающее USB-устройство объемом 32 Мбайт, смонтированное в каталоге /dev/sdb:

```
$ sudo fdisk /dev/sdb                Выполняет команду для управления
                                     разделами диска
Command (m for help): p              Отображает список существующих
                                     разделов (таких нет)
Disk /dev/sdb: 32 MB, 32112128 bytes
1 heads, 62 sectors/track, 1011 cylinders
Units = cylinders of 62 * 512 = 31744 bytes
Device Boot           Start  End         Blocks Id    System
Command (m for help): n              Создает новый раздел
Command action
e extended
p primary partition (1-4)
p                                    Делает данный раздел основным
Partition number (1-4): 1            Назначает разделу номер 1
First cylinder (2-1011, default 2): Enter
Using default value 2
Last cylinder or +size or +sizeM or +sizeK (2-1011, default 1011): Enter
Using default value 1011
Command (m for help): t              Определяет тип раздела
Selected partition 1
Hex code (type L to list codes): 8E       Код LVM-раздела
Changed system type of partition 1 to 8e (Linux LVM)
Command (m for help): p              Отображает информацию о новом разделе
Disk /dev/sdb: 32 MB, 32112128 bytes
1 heads, 62 sectors/track, 1011 cylinders
Units = cylinders of 62 * 512 = 31744 bytes
Device Boot     Start   End     Blocks      Id       System
/dev/sdb1       2       1011    31310       8e       Linux LVM
```

Прежде чем продолжать, убедитесь, что были произведены правильные изменения в правильном разделе! Если все верно, создайте новую таблицу разбиения:

```
Command (m for help): w

The partition table has been altered!
Calling ioctl() to re-read partition table.
Syncing disks.
```

Вернувшись в консоль, воспользуйтесь командой sfdisk для просмотра схемы разбиения жесткого диска:

```
$ sudo sfdisk -l /dev/sdb          Выводит список LVM-разделов
Disk /dev/sdb: 1011 cylinders, 1 heads, 62 sectors/track
Units = cylinders of 31744 bytes, blocks of 1024 bytes, counting from 0
   Device Boot     Start    End    #cyls   #blocks   Id     System
   /dev/sdb1       1        1010   101     31310     8e     Linux LVM
   /dev/sdb2       0        -      0       0         0      Empty
   /dev/sdb3       0        -      0       0         0      Empty
   /dev/sdb4       0        -      0       0         0      Empty
```

После этого назначьте раздел /dev/sdb1 в качестве нового физического тома LVM и воспользуйтесь командой pvs для просмотра информации о существующих LVM-томах:

```
$ sudo pvcreate /dev/sdb1          Делает носитель sdb1 физическим томом LVM
Physical volume "/dev/sdb1" successfully created
$ sudo pvs View physical LVM partitions
PV          VG          Fmt         Attr        PSize       PFree
/dev/sdb1 vgusb lvm2    a-          28.00M      20.00M
```

Теперь создайте группу томов vgusb, воспользовавшись командой vgcreate, и отобразите текущие группы томов:

```
$ sudo vgcreate vgusb /dev/sdb1           Создает группу томов vgusb
   Volume group "vgusb" successfully created
$ sudo vgs                                Отображает текущие группы томов
   VG         #PV    #LV    #SN    Attr        VSize Vfree
   Vgusb 1    0      0      wz--n- 28.00M28.00M
```

Для создания нового LVM-раздела размером 10 Мбайт из группы томов vgusb воспользуйтесь командой lvcreate. Затем с помощью команды lvs просмотрите информацию о логическом томе, а с помощью команды vgs определите объем изменившегося свободного места:

```
$ sudo lvcreate --size 10M --name lvm_u1 vgusb
   Rounding up size to full physical extent 12.00 MB
   Logical volume "lvm_u1" created
$ sudo lvs                Отображает информацию о логическом томе
   LV         VG          Attr        LSize OriginSnap% Move Log Copy%
   lvm_u1 vgusb -wi-a- 12.00M
$ sudo vgs                Показывает, что у вас есть 16 Мбайт свободного места
   VG         #PV         #LV    #SN    Attr        VSize     VFree
   Vgusb      1     1            0      wz--n-      28.00M    16.00M
```

Чтобы создать на LVM-разделе файловую систему ext3, выполните следующую команду:

```
$ sudo mkfs.ext3 /dev/mapper/vgusb-lvm_u1
mke2fs 1.38 (30-Jun-2005)
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
3072 inodes, 12288 blocks
614 blocks (5.00%) reserved for the super user
First data block=1
Maximum filesystem blocks=12582912
2 block groups
8192 blocks per group, 8192 fragments per group
1536 inodes per group
Superblock backups stored on blocks:
        8193
Writing inode tables: done
Creating journal (1024 blocks): done
Writing superblocks and filesystem accounting information: done
This filesystem will be automatically checked every 35 mounts or
180 days, whichever comes first. Use tune2fs -c or -i to override.
```

Теперь файловая система ext3 создана и том LVM готов к использованию.

Использование томов LVM

Чтобы приступить к использованию нового тома, представленного /dev/mapper/vgusb-lvm_u1, **создайте точку монтирования** (/mnt/u1) и **смонтируйте** этот том. Затем для проверки имеющегося свободного места воспользуйтесь командой df:

```
$ sudo mkdir /mnt/u1  Создает точку монтирования
$ sudo mount -t ext3 /dev/mapper/vgusb-lvm_u1 /mnt/u1 Монтирует том
$ df -m /mnt/u1             Проверяет обьем доступного дискового пространства
Filesystem 1M-blocks Used Available Use% Mounted on
/dev/mapper/vgusb-lvm_u1
12 2 10 11% /mnt/u1
```

На данном этапе файловая система содержит только каталог lost+found:

```
$ ls /mnt/u1
lost+found
```

Скопируйте крупный файл в файловую систему. Например, скопируйте один из файлов ядра из каталога /boot в /mnt/u1:

```
$ cp /boot/vmlinuz-* /mnt/u1/      Копирует крупный файл в /mnt/u1
$ df -m /mnt/u1                    Показывает, что каталог /mnt/u
                                   имеет обьем 4 Мбайт
Filesystem 1M-blocks Used Available Use% Mounted on
/dev/mapper/vgusb-lvm_u1
12 4 9 27% /mnt/u1
```

Сверьте информацию о скопированном файле с его md5sum и сохраните полученную контрольную сумму для дальнейшего использования:

```
$ md5sum /mnt/u1/vmlinuz-2.6.20-1.2316.fc5      Сверяет информацию с md5sum
8d0dc0347d36ebd3f6f2b49047e1f525 /mnt/u1/vmlinuz-2.6.20-1.2316.fc5
```

Увеличение тома LVM

Допустим, вы исчерпали доступное свободное место и вам необходимо расширить объем тома LVM. Чтобы это сделать, демонтируйте том и воспользуйтесь командой lvresize. После этого с помощью команды e2fsck проверьте файловую систему и выполните команду resize2fs, чтобы изменить размер файловой системы ext3 данного тома:

```
$ sudo umount /mnt/u1                            Демонтирует том
$ sudo lvresize --size 16M /dev/vgusb/lvm_u1     Изменяет размер тома
Extending logical volume lvm_u1 to 16.00 MB
Logical volume lvm_u1 successfully resized
$ sudo e2fsck -f /dev/vgusb/lvm_u1
e2fsck 1.40 (12-Jul-2007)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/vgusb/lvm_u1: 12/3072 files (25.0% non-contiguous), 3379/12288 blocks
$ sudo resize2fs /dev/vgusb/lvm_u1 16M   Изменяет размер файловой системы
resize2fs 1.38 (30-Jun-2005)
Resizing the filesystem on /dev/vgusb/lvm_u1 to 16384 (1k) blocks.
The filesystem on /dev/vgusb/lvm_u1 is now 16384 blocks long.
```

В приведенном примере и размер тома, и размер файловой системы был увеличен до 16 Мбайт.

После этого снова смонтируйте том, проверьте объем доступного дискового пространства и сверьте его с созданным ранее md5sum:

```
$ sudo mount -t ext3 /dev/mapper/vgusb-lvm_u1 /mnt/u1 Заново монтирует том
$ df -m /mnt/u1       Показывает, что использовано 4 Мбайт из 16 Мбайт
Filesystem      1M-blocks    Used Available     Use%  Mounted on
/dev/mapper/vgusb-lvm_u1
                16           4      13          20%   /mnt/u1
$ md5sum /mnt/u1/vmlinuz-2.6.20-1.2316.fc5      Заново проверяет md5sum
8d0dc0347d36ebd3f6f2b49047e1f525 /mnt/u1/vmlinuz-2.6.20-1.2316.fc5
```

Теперь смонтированный том занимает 16 Мбайт вместо 10 Мбайт.

Уменьшение размера тома LVM

Вы также можете использовать команду lvresize, если хотите **уменьшить объем существующего LVM-тома**. Как и в предыдущем случае, перед изменением раз-

мера тома размонтируйте его, а затем выполните команды e2fsck (чтобы проверить файловую систему) и resize2fs (для уменьшения размера тома):

```
$ sudo umount /mnt/u1
$ sudo e2fsck -f /dev/vgusb/lvm_u1
fsck 1.38 (30-Jun-2005)
e2fsck 1.38 (30-Jun-2005)
The filesystem size (according to the superblock) is 16384 blocks
The physical size of the device is 8192 blocks
Pass 1: Checking inodes, blocks, and sizes
...
/dev/vgusb/lvm_u1: 12/3072 files (8.3% non-continguous,3531/16384 blocks
$ sudo resize2fs /dev/vgusb/lvm_u1 12M   Изменяет размер файловой системы
resize2fs 1.38 (30-Jun-2005)
Resizing the filesystem on /dev/vgusb/lvm_u1 to 12288 (1k) blocks.
The filesystem on /dev/vgusb/lvm_u1 is now 12288 blocks long.
$ sudo lvresize --size 12M /dev/vgusb/lvm_u1
 WARNING: Reducing active logical volume to 12.00 MB
 THIS MAY DESTROY YOUR DATA (filesystem etc.)
Do you really want to reduce lvm_u1? [y/n]: y
 Reducing logical volume lvm_u1 to 8.00 MB
 Logical volume lvm_u1 successfully resized
$ sudo mount -t ext3 /dev/mapper/vgusb-lvm_u1 /mnt/u1    Заново монтирует том
$ df -m /mnt/u1       Показывает, что использовано 4 Мбайт из 12 Мбайт
Filesystem            1M-blocks   Used        Available    Use%     Mounted on
/dev/mapper/vgusb-lvm_u1
                      12          4           9            20%   /mnt/u1
```

Теперь смонтированный том занимает 12 Мбайт вместо 16 Мбайт.

Удаление логических томов и групп LVM

Чтобы **удалить логический LVM-том из группы томов**, используйте команду lvremove:

```
$ sudo lvremove /dev/vgusb/lvm_u1
Do you really want to remove active logical volume "lvm_u1"? [y/n]: y
   Logical volume "lvm_u1" successfully removed
```

Для удаления целой группы LVM-томов используйте команду vgremove:

```
$ sudo vgremove vgusb
   Volume group "vgusb" successfully removed
```

Существует гораздо больше способов работы с LVM, ознакомиться с которыми вы можете в статье «LVM HOWTO» (http://tldp.org/HOWTO/LVM-HOWTO/).

Резюме

Создание файловых систем в Linux и управление ими является очень важной частью системного администрирования. Linux поддерживает некоторые классические

типы файловых систем (ext2, ext3, reiserfs и др.), а также позволяет создавать и управлять файловыми системами Windows (VFAT, NTFS и т. д.) и традиционными и специальными типами файловых систем Linux и UNIX (minix, jfs и xfs).

С помощью таких команд, как fdisk и parted, вы можете разбивать пространство жесткого диска. Среди приложений, предназначенных для работы с файловыми системами, можно отметить утилиты для их создания (mkfs), просмотра и изменения атрибутов файловой системы (tune2fs и mpe2fs), монтирования и демонтирования файловых систем (mount и umount) и проверки на наличие ошибок и поврежденных секторов (badblocks и fsck). Чтобы получить информацию о занимаемом файлами и каталогами пространстве файловой системы, используйте команды df и du.

8 Создание резервных копий и работа со съемными носителями

Раньше резервное копирование данных в Linux, как правило, осуществлялось путем сжатия и архивирования данных с последующей записью полученного архива на ленту. В последние годы количество утилит для архивирования, способов сжатия и видов копируемых данных существенно возросло. Архивирование с использованием ленты было вытеснено другими способами резервного копирования, такими как сетевое резервное копирование, копирование на другие жесткие диски, а также на компакт-диски, DVD или другие недорогие съемные носители.

В данной главе подробно рассматриваются приложения, используемые для резервного копирования и восстановления важных данных. Первая часть главы посвящена использованию таких основных приложений резервного копирования, как tar, gzip и rsync.

Резервное копирование данных в архивы

Если вы раньше работали в Windows, то вам, скорее всего, приходилось использовать такие приложения, как WinZip или PKZIP, сжимающие группы файлов в один. В Linux для помещения групп файлов в один архив (`tar`) и сжатия этого архива для последующего эффективного хранения (`gzip`, `bzip2` и `lzop`) предлагаются разные инструменты, однако с помощью дополнительных параметров команды `tar` вы можете выполнять эти две операции совместно.

Создание резервных архивов с помощью инструмента tar

Команда `tar`, выступающая в качестве ленточного архиватора, относится к приложениям ранних систем UNIX. Хотя первоначальным носителем, на который записывалась информация с помощью `tar`, была еще магнитная лента, `tar` и сегодня используется для создания архивов, которые впоследствии могут быть записаны на различные носители.

Команда `tar` способна выполнять большое количество функций, что отражено в десятках ее параметров. Однако в основном `tar` используется для создания

резервных архивов (-c), извлечения файлов из архива (-x), выявления различий между архивами (-d) и обновления файлов в архиве (-u). Кроме того, с помощью этой команды вы можете добавлять файлы в существующие архивы (-r или -A), удалять файлы из них (-d), а также отображать содержимое архива (-t).

ПРИМЕЧАНИЕ

Хотя команда tar входит в состав практически всех систем UNIX и Linux, на многих системах она используется по-разному. Например, операционная система Solaris не поддерживает использование параметра -z для управления архивами TAR в формате gzip, зато команда Star (ess-tar) поддерживает доступ к контрольным спискам (ACLs) и меткам (для обеспечения расширенных прав доступа при использовании Samba).

В процессе создания TAR-архива вы можете добавлять параметры для сжатия полученного архива. Например, для сжатия архива в формат bzip2 укажите параметр -j, а для сжатия в формат gzip — параметр -z. По определению, обычные файлы tar имеют расширение TAR, в то время как сжатые архивы — TAR.BZ2 (сжат с помощью bzip2) или TAR.GZ (сжат с помощью gzip). Если же вы сожмете файл вручную с помощью утилиты lzop (www.lzop.org), то полученный архив будет иметь расширение TAR.LZO.

Файлы TAR, помимо использования для резервного копирования, являются популярным способом распространения исходных кодов и бинарных файлов проектами по разработке программного обеспечения. Именно поэтому вы вправе рассчитывать, что на любой системе Linux и UNIX будут приложения, необходимые для работы с файлами tar.

ПРИМЕЧАНИЕ

Особенность работы с командой tar связана с тем, что она была создана еще до возникновения стандартов использования параметров. Хотя вы можете добавлять параметры для tar с помощью дефиса, это не всегда является обязательным условием. Поэтому вы вполне можете встретить команду tar, например tar xvf, без дефиса, обозначающего ввод параметра.

В качестве классического примера использования команды tar может служить комбинирование устаревших параметров и каналов для сжатия информации:

```
$ tar c *.txt | gzip -c > myfiles.tar.gz       Создает архив, сжимает его
                                               и возвращает
```

Приведенный пример, который вы можете найти в документации к старым системам UNIX, демонстрирует выполнение стандартного двухшагового процесса. Команда tar создает в текущем каталоге архив из всех текстовых TXT-файлов (c), затем направляет выход команде gzip и выводит результат на stdout (-c), а уже после этого направляет информацию в файл myfiles.tar.gz. Стоит отметить, что tar является одной из немногих команд, которая не требует использования знака дефиса (-) для добавления параметра.

Новые версии tar, используемые на современных системах Linux, способны **создавать архивы и сжимать информацию в рамках одной операции**:

```
$ tar czf myfiles.tar.gz *.txt        Из TXT-файлов создает файл tar,
                                      сжатый с помощью gzip
```

```
$ tar czvf myfiles.tar.gz *.txt       При создании архива отображает более
                                      подробную информацию
textfile1.txt
textfile2.txt
```

Обратите внимание, что в приведенных примерах имя архива (myfiles.tar.gz) должно следовать сразу за параметром f команды tar (он определяет имя будущего архива) — иначе выход tar будет направлен в stdout (другими словами, на экран вашего монитора). Параметр z указывает, что для сжатия необходимо использовать программу gzip, а параметр v позволяет отображать более подробную информацию о процессе сжатия.

Кроме того, с помощью команды tar (и при желании gunzip) вы можете **распаковывать архивы**. Выполнить эту операцию можно в один или два этапа:

```
$ gunzip -c myfiles.tar.gz | tar x           Распаковывает архивы ZIP и TAR
$ gunzip myfiles.tar.gz ; tar xf myfiles.tar  Распаковывает архивы ZIP и TAR
```

Чтобы выполнить ту же операцию в одно действие, введите следующую команду:

```
$ tar xzvf myfiles.tar.gz textfile1.txt textfile2.txt
```

В результате ее выполнения будут распакованы и скопированы в текущий каталог заархивированные ранее TXT-файлы. Параметр x используется для извлечения файлов, z — для распаковывания, параметр v расширяет объем отображаемой информации, а параметр f указывает, что следующий параметр является именем обрабатываемого архива (myfiles.tar.gz).

спользование приложений для сжатия

Сжатие является важным этапом при резервном копировании файлов. Сжатые файлы занимают меньше места на носителях (компакт-диске, DVD, магнитной ленте и т. д.) или сервере, используемом для хранения резервных копий файлов. Кроме того, для копирования таких файлов на носитель или передачи их по сети требуется меньше времени.

Сжатие, с одной стороны, сохраняет много пространства на носителях и сокращает время передачи файлов на сервер, но, с другой стороны, может значительно увеличить нагрузку на центральный процессор. Для примера вы можете ознакомиться с процедурой аппаратного сжатия данных с использованием ленты (www.amanda.org/docs/faq.html#id346016).

В приведенных выше примерах вместе с tar использовалась команда gzip. Тем не менее tar может работать и со многими другими приложениями для сжатия. В Ubuntu tar работает с утилитами gzip и bzip2. Третьей включенной в данное руководство утилитой для сжатия является команда lzop, которая может использоваться с tar несколько иначе, чем остальные две. Если выстроить эти приложения в порядке от самого быстрого, но обеспечивающего наименьшую степень сжатия к самому медленному, но обеспечивающему наилучшее сжатие, то получится следующая последовательность: lzop, gzip и bzip2.

Если вы архивируете и сжимаете большое количество информации, это может занять значительное время. В этом случае вам стоит иметь в виду, что команда bzip2 тратит на сжатие примерно в десять раз больше времени, чем lzop, а сжимает лучше только в два раза. Однако каждая команда позволяет использовать различные уровни сжатия, что несколько уравнивает соотношения степени сжатия и времени, затрачиваемого на него.

Чтобы использовать команду tar вместе с bzip2, используйте параметр -j:

```
$ tar cjvf myfiles.tar.bz2 *.txt   Создает архив и сжимает его с помощью bzip2
```

Используя параметр -j, вы можете и **распаковывать сжатые с помощью bzip2 файлы** (так же, как при использовании параметра -x с командой tar):

```
$ tar xjvf myfiles.tar.bz2  Извлекает файлы, распаковая их с помощью bzip2
```

Утилита lzop несколько меньше совместима с tar. Для ее использования вам потребуется установить пакет lzop. Чтобы **выполнить сжатие с помощью команды lzop**, воспользуйтесь параметром --use-compress-program:

```
$ sudo apt-get install lzop
$ tar --use-compress-program=lzop -cf myfiles.tar.lzo *.txt
$ tar --use-compress-program=lzop -xf myfiles.tar.lzo
```

В данных примерах команда использует синтаксис, обратный старому синтаксису команды tar, располагая параметр перед командой. В остальных примерах мы использовали современный синтаксис команды tar, не требующий использования параметра.

ПРИМЕЧАНИЕ

Вы также можете столкнуться со сжатыми файлами в формате RAR. Этот формат популярен в сетях с равноправными узлами и защищен авторскими правами, поэтому не имеет какого-либо широко распространенного в Linux приложения для работы с файлами RAR. В Ubuntu, чтобы получить возможность работать с командами для файлов в формате RAR, установите программные пакеты unrar и rar.

Команда gzip

Как уже было отмечено, вы **можете любую команду сжатия использовать отдельно** (в отличие от команды tar). Ниже приведено несколько примеров использования команды gzip, позволяющих создавать сжатые файлы gzip и работать с ними:

```
$ gzip myfile          Сжимает myfile и переименовывает его в myfile.gz
```

Следующая команда выполняет те же действия, но отображает более подробную информацию о ходе выполнения:

```
$ gzip -v myfile       Сжимает myfile и отображает более подробную информацию
myfile: 86.0% -- replaced with myfile.gz
$ gzip -tv myfile.gz   Проверяет целостность файла gzip
myfile.gz: OK
$ gzip -lv myfile.gz   Отображает подробную информацию о файле gzip
method  crc       date    time  compressed  uncompressed  ratio
   uncompressed_name
defla   0f27d9e4  Jul 10  04:48 46785       334045        86.0%
   myfile
```

Для **сжатия всех файлов в каталоге** воспользуйтесь любой из следующих команд:

```
$ gzip -rv mydir           Сжимает все файлы в каталоге
mydir/file1: 39.1% -- replaced with mydir/file1.gz
mydir/file2: 39.5% -- replaced with mydir/file2.gz
$ gzip -1 myfile           Обеспечивает наибольшую скорость,
                           но наименьшую степень сжатия
$ gzip -9 myfile           Обеспечивает наименьшую скорость,
                           но наибольшую степень сжатия
```

Для определения уровня сжатия добавьте дефис перед соответствующей цифрой (от 1 до 9). Как следует из приведенного выше примера, число -1 соответствует наиболее высокой скорости и наименьшей степени компрессии, а число -9 — наименьшей скорости и наибольшей степени компрессии. Уровнем по умолчанию для gzip является 6. Команда lzop использует меньшее количество уровней: 1, 3 (по умолчанию), 7, 8 и 9. Уровни сжатия команды bzip2 действуют иначе.

Для **распаковки файла gzip** используйте команду gunzip и один из следующих примеров:

```
$ gunzip -v myfile.gz         Распаковывает файл myfile.gz
                              и переименовывает его в myfile
myfile.gz:      86.0% -- replaced with myfile
$ gzip -dv myfile.gz          То же самое
```

Хотя приведенные примеры демонстрируют процесс упаковки обычных файлов, те же параметры могут использоваться и для сжатия файлов tar.

Команда bzip2

Команда bzip2 позволяет достигать максимальной, по сравнению со всеми остальными приводимыми в данной главе командами, степени компрессии. Ниже представлены некоторые примеры использования команды bzip2:

```
$ bzip2 myfile                Сжимает файл и переименовывает его в myfile.bz2
$ bzip2 -v myfile             То же, но выводит более подробную информацию
 myfile: 9.529:1, 0.840 bits/byte, 89.51% saved, 334045 in, 35056 out.
$ bunzip2 myfile.bz2          Распаковывает файл и переименовывает его в myfile
$ bzip2 -d myfile.bz2         То же
$ bunzip2 -v myfile.bz2       То же, но выводит более подробную информацию
 myfile.bz2: done
```

Команда lzop

Использование команды lzop отличается от использования gzip и bzip2. Данная команда лучше подходит для случаев, при которых скорость сжатия важнее его степени. Когда lzop сжимает содержимое файла, она оставляет исходный файл нетронутым (если вы не используете параметр -U), однако создает новый файл с расширением LZO. Вы можете воспользоваться любым из следующих примеров **использования команды lzop для сжатия файла** myfile:

```
$ lzop -v myfile       Сохраняет myfile и создает сжатый myfile.lzo
compressing myfile into myfile.lzo
$ lzop -U myfile       Удаляет myfile, но создает сжатый myfile.lzo
```

Чтобы после создания файла myfile.lzo **проверить, отобразить или распаковать его**, воспользуйтесь любой из следующих команд:

```
$ lzop -t myfile.lzo          Проверяет целостность сжатого файла
$ lzop --info myfile.lzo      Отображает заголовок каждого вложенного файла
$ lzop -l myfile.lzo          Отображает информацию о сжатии для каждого
                              вложенного файла
method    compressed   uncompr.     ratio      uncompressed_name
LZO1X-1   59008        99468        59.3%      myfile
$ lzop --ls myfile.lzo        Отображает содержимое сжатого файла
                              (аналогично команде ls -l)
$ cat myfile | lzop > x.lzo   Сжимает standin и направляет его в stdout
$ lzop -dv myfile.lzo         Сохраняет файл myfile.lzo и распаковывает его
                              в myfile
```

В отличие от gzip и bzip2, команда lzop не имеет соответствующей команды для распаковки архивов. Для распаковки файла просто добавьте к команде lzop параметр -d. Если передать команде lzop список из файлов и имен каталогов, то она сожмет все файлы, пропустив при этом каталоги. В полученном запакованном файле сохранятся все параметры исходных элементов: имена, права доступа и подписи даты и времени.

Просмотр, объединение и добавление файлов в архивы tar

До сих пор с помощью tar мы только создавали и распаковывали архивы, однако данная команда поддерживает также параметры, позволяющие просматривать содержимое архивов, объединять их, добавлять файлы в существующие архивы и удалять файлы из архива.

Чтобы **просмотреть содержимое архива**, воспользуйтесь параметром -t:

```
$ tar tvf myfiles.tar         Отображает список вложенных в TAR-архив файлов
-rw-r--r-- root/root 9584 2007-07-05 11:20:33 textfile1.txt
-rw-r--r-- root/root 9584 2007-07-09 10:23:44 textfile2.txt
$ tar tzvf myfiles.tgz        Отображает файлы архива gzip
```

Если архив сжат с помощью команды lzop и назван myfile.tar.lzo, то вы можете **отобразить содержимое этого tar/lzop-файла** следующим образом:

```
$ tar --use-compress-program=lzop -tf myfiles.tar.lzo    Отображает содержимое
                                                         архива LZO
```

Для **объединения одного файла tar с другим** предназначен параметр -A. Следующая команда добавляет содержимое archive2.tar к архиву archive1.tar:

```
$ tar -Af archive1.tar archive2.tar
```

Чтобы **добавить один или несколько файлов** в существующий архив, воспользуйтесь параметром -r. В следующем примере файл myfile добавляется в архив archive.tar:

```
$ tar rvf archive.tar myfileДобавляет файл в архив tar
```

С помощью специальных символов вы также можете **добавить несколько файлов** в архив:

```
$ tar rvf archive.tar *.txt        Добавляет несколько файлов в архив tar
```

Удаление файлов из архивов tar

Если у вас на жестком диске имеется файл архива tar, то вы при необходимости можете удалить некоторые файлы из этого архива. Отметим, что данный способ нельзя использовать для удаления файлов архивов tar, записанных на магнитной ленте. Ниже приведен пример **удаления файлов из архива tar**:

```
$ tar --delete file1.txt -f myfile.tar   Удаляет file1.txt из myfile.tar
```

Резервное копирование файлов через сеть

После выполнения резервного копирования файлов и помещения их в архив что вы будете делать с полученным архивом? Первой причиной, по которой необходимо сделать резервную копию этого архива, является вероятность, например, выхода из строя жесткого диска, в случае чего вам потребуется восстанавливать файлы из резервной копии. Сохранить резервные копии можно следующими способами:

- скопировав резервные копии на съемные носители, например ленту, компакт-диск или DVD (как было описано ранее в этой главе);
- скопировав их на другой компьютер сети.

Скорость и надежность сетей, а также низкая стоимость жестких дисков большой вместимости и безопасность, связанная с перемещением данных за пределы рабочего места, делают популярным резервное копирование через сеть. Для выполнения эффективного и безопасного резервного копирования личных данных или данных небольшого офиса может быть достаточно сочетания нескольких простых команд. Подобный подход очень хорошо характеризует философию UNIX, заключающуюся в объединении нескольких простых программ с целью выполнения ими более сложных задач.

Хотя практически любая команда, способная копировать файлы по сети, может осуществить резервное копирование на удаленный компьютер, некоторые утилиты особенно хорошо справляются с этой задачей. С помощью инструментов OpenSSH, таких как ssh и scp, вы можете настроить безопасную и зашифрованную передачу резервных архивов.

Приложения наподобие команды `rsync` могут сохранять ресурсы, копируя только файлы (или части файлов), измененные со времени последнего резервного копирования. С помощью таких команд, как `unison`, вы можете осуществлять резервное копирование файлов как в Windows, так и в операционных системах Linux.

В следующих подразделах описываются способы резервного копирования данных по сети на другие компьютеры.

ПРИМЕЧАНИЕ

Кроме того, приложением, которое может вас заинтересовать, является команда rsnapshot (yum install rsnapshot) (www.rsnapshot.org), которая может работать совместно с rsync и создавать ежечасные, ежедневные, еженедельные и ежемесячные (параметр времени настраивается) копии файловой системы. Для сохранения копий файловой системы используются жесткие ссылки, которые впоследствии можно синхронизировать с измененными файлами.

Установить данное приложение можно с помощью следующих команд:

```
$ sudo apt-get install rsnapshot
$ sudo apt-get install sshfs
```

Резервное копирование архивов tar через ssh

Пакет OpenSSH (www.openssh.org) предоставляет набор приложений для безопасного удаленного входа в систему, удаленного выполнения и удаленного копирования файлов через сетевые интерфейсы. Настроив два компьютера на совместное использование шифров-ключей, вы сможете обмениваться файлами между этими машинами без необходимости вводить пароль при каждой попытке передачи данных, что позволит вам создавать коды для автоматического резервного копирования данных из SSH-клиента на SSH-сервер.

Используя команды OpenSSH и работая на центральной системе Linux, вы можете **собирать резервные копии данных с машин-клиентов**. Следующая команда выполняет команду tar на удаленном клиенте (для архивирования и сжатия файлов), направляет поток данных на стандартный выход и использует команду ssh для локального размещения резервных копий данных посредством tar:

```
$ mkdir mybackup ; cd mybackup
$ ssh francois@server1 'tar cf - myfile*' | tar xvf -
francois@server1's password: ******
myfile1
myfile2
```

В приведенном примере все файлы, начинающиеся с myfile, копируются из рабочего каталога пользователя francois на server1 и помещаются в текущий каталог. Заметьте, что с левой стороны канала создается архив, а с правой стороны файлы из архива перемещаются в текущий каталог (запомните, что ssh переписывает локальные файлы, если таковые существуют, и именно поэтому в примере мы создали пустой каталог).

Чтобы выполнить обратное действие и скопировать файлы **с локальной системы на удаленную**, необходимо сначала выполнить команду tar, а затем для помещения файлов в выбранный каталог на удаленном компьютере использовать команду cd:

```
$ tar cf - myfile* | ssh francois@server1 \
 'cd /home/francois/myfolder; tar xvf -'
francois@server1's password: ******
myfile1
myfile2
```

В следующем примере мы не распаковываем файлы tar на передающей стороне, а **записываем результаты в файлы TGZ**:

```
$ ssh francois@server1 'tar czf - myfile*' | cat > myfiles.tgz
$ tar cvzf - myfile* | ssh francois@server1 'cat > myfiles.tgz'
```

В первом примере файлы из рабочего каталога пользователя francois, хранящиеся на server1 и начинающиеся на myfile, запаковываются в архив TAR, который затем сжимается, после чего уже запакованные файлы направляются в файл myfiles.tgz на локальной системе. Во втором примере выполняется обратное действие: из локального каталога выбираются все файлы, начинающиеся на myfile, и направляются в файл myfiles.tgz, находящийся на удаленной системе.

Приведенные в этих примерах команды удобно использовать для копирования файлов через сеть. Помимо сжатия, они позволяют использовать на ваш выбор различные параметры tar, такие как инкрементные параметры резервного копирования.

Резервное копирование файлов с помощью rsync

Команда rsync поддерживает большое количество параметров для резервного копирования. Уникальной данную команду делает алгоритм, позволяющий сверять на основании контрольных сумм небольшими блоками локальные и удаленные файлы и передавать только отличающиеся блоки. Этот алгоритм настолько эффективен, что используется и в других программах резервного копирования.

Команда rsync может использоваться перед удаленным командным процессором (ssh) или посредством запуска демона rsyncd в конце сервера. В следующем примере rsync используется через консоль ssh для получения зеркала каталога:

```
$ rsync -avz --delete chris@server1:/home/chris/pics/ chrispics/
```

Приведенная команда предназначена для отражения в локальной системе структуры удаленного каталога (/home/chris/pics/). Параметр -a выполняет команду в режиме архивации (рекурсивного копирования всех файлов из удаленного каталога), параметр -z сжимает файлы, а -v детализирует возвращаемую информацию. Параметр --delete указывает команде rsync на необходимость удалять любые файлы на локальной системе, которые больше не существуют на удаленной.

Для непрерывного пошагового резервного копирования с помощью команды rsync выполните следующую команду:

```
# mkdir /var/backups

# rsync --delete -backup                                  \
  --backup-dir=/var/backups/backup-`date +%A`            \
  -avz chris@server1:/home/chris/Personal/               \
  /var/backups/current-backup/
```

При запуске приведенной команды все файлы из каталога /home/chris/Personal на удаленном компьютере server1 копируются в локальный каталог /var/backups/,

предназначенный для хранения текущих резервных копий. Все изменяющиеся файлы копируются в каталог с соответствующим дню недели названием, например /var/backups/backup-Monday. По окончании недели будет создано семь каталогов, отражающих все произведенные за это время изменения.

Другой хитростью для выполнения циклического резервного копирования является использование вместо многочисленных копий файлов жестких ссылок. Этот процесс осуществляется в два этапа и заключается в циклическом сдвиге файлов с последующим выполнением команды rsync:

```
# rm -rf /var/backups/backup-old/
# mv /var/backups/backup-current/ /var/backups/backup-old/
# rsync --delete --link-dest=/var/backups/backup-old -avz \
  chris@server1:/home/chris/Personal/ /var/backups/backup-current/
```

В предыдущем примере текущий резервный каталог заменяется предыдущим посредством удаления резервных копий прошлой недели и замены их резервными копиями текущей недели. Если любой из файлов, скопированных из личного удаленного каталога и расположенных на компьютере server1, существовал при предыдущем резервном копировании, то при запуске посредством команды rsync с параметром --link-dest нового резервного копирования будет создана жесткая ссылка, связывающая файлы в текущем и старом резервных каталогах.

С помощью жестких ссылок между файлами в старом и текущем резервных каталогах вы можете сэкономить много места на диске. Так, если у вас в обоих каталогах есть файл file1.txt, то, отобразив файлы inodes следующим образом, вы можете убедиться, что оба файла являются одним физическим файлом:

```
$ ls -i /var/backups/backup*/file1.txt
260761 /var/backups/backup-current/file1.txt
260761 /var/backups/backup-old/file1.txt
```

Команда unison

Хотя команда rsync и является удобной для осуществления резервного копирования с одного компьютера на другой, однако это предполагает, что данные изменяются только на компьютере, с которого осуществляется резервное копирование. В случае же если у вас есть два компьютера и они оба изменяют один файл, а вам необходимо синхронизировать эти файлы, придется использовать приложение unison.

Часто возникает необходимость работать с одними и теми же документами одновременно на ноутбуке и настольном компьютере. На этих компьютерах могут даже быть установлены разные операционные системы, однако, поскольку unison является межплатформенным приложением, оно позволяет синхронизировать файлы, находящиеся и на системах Linux, и на системах Windows. Чтобы использовать приложение unison в Ubuntu, вам потребуется установить одноименный программный пакет (выполните из учетной записи команду sudo и apt-get install unison).

С помощью unison вы сможете создать учетные записи двух суперпользователей, тем самым обеспечив два пути синхронизации. Эти учетные записи могут быть как локальными, так и удаленными:

```
$ unison /home/francois ssh://francois@server1//home/fcaen
$ unison /home/francois /mnt/backups/francois-homedir
```

ПРИМЕЧАНИЕ

Убедитесь, что на обоих компьютерах установлена одинаковая версия unison.

Для осуществления резервного копирования команда unison позволяет использовать как графические, так и консольные приложения. По умолчанию используется графическая версия. Тем не менее, если у вас не запущен оконный менеджер или если у вас unison запущен без графической оболочки, то это может вызвать некоторые ошибки в работе программы. Чтобы **запустить unison в режиме командной строки**, воспользуйтесь параметром -ui text:

```
$ unison /home/francois ssh://francois@server1//home/fcaen -ui text
Contacting server...
francois@server1's password:
Looking for changes
   Waiting for changes from server
Reconciling changes
local    server1
newfile ---->  memo.txt [f] y
Propagating updates
   ...
```

После этого утилита utility сравнит две учетные записи суперпользователей и для каждого зафиксируемого изменения попросит определить вид требуемого действия. В приведенном выше примере на локальной системе был обнаружен новый файл memo.txt. От пользователя требуется определить, хочет ли он продолжать выполнение обновления (в этом случае файл memo.txt будет скопирован с локального компьютера на server1). Чтобы принять обновления, введите y.

Вы также можете добавить к команде параметр -auto, чтобы утилита **выполняла предусмотренные по умолчанию действия, не запрашивая подтверждения пользователя на их выполнение**:

```
$ unison /home/francois ssh://francois@server1//home/fcaen -auto
```

Более подробную информацию вы можете получить, ознакомившись с MAN-страницей, посвященной unison. Кроме того, воспользовавшись параметром -help, вы можете просмотреть полный список используемых командой unison параметров, а с помощью параметра -doc all — постранично отобразить руководство по unison:

```
$ unison -help                Отображает список параметров unison
$ unison -doc all | less      Выводит руководство по unison
```

Если вам часто приходится синхронизировать учетные записи суперпользователя, **создайте профиль**, основанный на заготовленных настройках. В графическом

режиме это можно сделать на экране по умолчанию. Профили же хранятся в каталоге ~/.unison/ в текстовых PRF-файлах, которые могут быть совсем простыми:

```
root = /home/francois
root = ssh://francois@server1//home/fcaen
```

Если настройки хранятся в профиле fc-home.prf, то можете активизировать его с помощью следующей команды:

```
$ unison fc-home
```

Резервное копирование данных на съемные носители

Вместительность компакт-дисков и DVD, а также низкая цена на эти виды носителей сделали их популярными для резервного копирования данных с компьютеров. Используя приложения, входящие в большинство операционных систем Linux, вы можете собирать файлы в виде образов для их последующего резервного копирования на компакт-диски или DVD.

Приложения командной строки, такие как mkisofs (создает и записывает образы на компакт-диски) и cdrecord (создает и записывает образы на компакт-диски и DVD), стали наиболее популярными для осуществления резервного копирования на компакт-диски и DVD. На сегодняшний день существует множество графических вариантов этих приложений, которые вы также можете использовать в качестве утилит для резервного копирования. Например, среди приложений GUI для подготовки и записи CD/DVD можно назвать K3b (записывает компакт-диски и DVD в KDE) и Nautilus (файловый менеджер для GNOME с возможностью записи компакт-дисков). Среди других GUI-приложений, предназначенных для записи компакт-дисков, можно отметить gcombust, X-CD-Roast и graveman.

Команды для создания образов файловых систем и их последующей записи и хранения на компакт-дисках или DVD описаны далее.

Создание резервных образов

Большинство компакт-дисков и DVD с данными, поскольку они создаются с использованием стандарта ISO9660, применяющегося для форматирования информации на этих дисках, читаются и в операционных системах Windows, и в операционных системах Linux. Поскольку большинство современных операционных систем вынуждено хранить большее количество информации о файлах и каталогах, чем включает в себя стандарт ISO9660, для этого стандарта с целью увеличения максимального объема хранимой информации были разработаны дополнительные расширения.

Используя команду mkisofs, вы можете осуществлять резервное копирование файлов и структуры каталогов из любой точки файловой системы Linux, а затем создавать образ ISO9660. Этот образ может включать в себя следующие типы расширений.

- **Общий протокол использования системы (SUSP)** — это записи, отмеченные в протоколе обмена Rock Ridge. Записи SUSP могут содержать UNIX-подобные атрибуты, такие как права собственности, поддержка длинных имен файлов и специальные файлы (такие как идентификаторы накопителя и символьные ссылки).
- **Joliet** — в каталоге с записями Joliet записываются длинные имена файлов, чтобы они могли использоваться в Windows.
- **Иерархическая файловая система (HFS)** — расширение, которое позволяет образу ISO восприниматься в виде файловой системы HFS, являющейся исходной для компьютеров Macintosh. Аналогичным образом благодаря использованию данной файловой системы могут быть добавлены ветви данных и ресурсов, чтобы они были доступны для чтения на компьютерах Mac.

При создании файла-образа ISO необходимо учитывать, в какой именно системе в конечном итоге будет осуществляться доступ к резервируемым с помощью команды `mkisofs` файлам (Linux, Windows или Mac). Созданный образ может использоваться разными способами, однако чаще всего он используется для записи на компакт-диск или DVD.

Помимо того, что приложение mkisofs может быть использовано для полного или частичного создания файловых систем Linux с целью их последующего использования на съемных носителях, оно позволяет создавать Live CD/DVD. Выполнить это можно, добавив в образ загрузочную информацию, с помощью которой можно загрузить ядро Linux или другой операционной системы, не используя жесткий диск компьютера.

ПРИМЕЧАНИЕ

Хотя mkisofs все еще может использоваться на Ubuntu, она все чаще заменяется командой genisoimage. Она является производной от mkisofs, которая была частью программного пакета cdrtools (http://cdrecord.berlios.de). Развитие genisoimage выполнялось в рамках проекта cdrkit (www.cdrkit.org).

Поскольку большинство пользователей Linux хранят свои личные файлы в рабочих каталогах, наиболее распространенным способом использования команды `mkisofs` является резервное копирование всей информации, хранящейся в дереве каталогов ниже каталога `/home`.

Ниже представлено несколько примеров использования команды `mkisofs` для **создания образа ISO из всех файлов и каталогов, расположенных ниже каталога** `/home`:

```
$ cd /tmp
$ sudo mkisofs -o home.iso /home                Создает базовый образ ISO9660
$ sudo mkisofs -o home2.iso -J -R /home         Добавляет расширения
                                                Joliet Rock Ridge
$ sudo mkisofs -o home3.iso -J -R -hfs /home    Добавляет расширения HFS
```

При выполнении последней команды вы увидите следующее предупреждение:

```
genisoimage: Warning: no Apple/Unix files will be decoded/mapped
```

В каждом из трех приведенных выше примеров все файлы и каталоги, расположенные ниже каталога `/home`, добавляются в образ ISO (`home.iso`). В первом примере

не использовалось расширений, поэтому имена всех файлов конвертировались в стиле DOS (формат 8.3). Во втором примере использовались расширения Joliet и Rock Ridge, поэтому в системе Linux или Windows имена файлов и права доступа к ним будут выглядеть так же, как и в исходной системе Linux. В последнем примере создается образ, читаемый файловой системой Mac.

ПРИМЕЧАНИЕ

Операционная система Mac OS X также способна определять расширения Rock Ridge и Joliet.

С помощью следующих команд вы можете добавлять к образу различные источники:

```
$ mkisofs -o home.iso -R -J music/ docs/ \   Добавляет несколько
                                             каталогов/файлов
 chris.pdf /var/spool/mail
$ mkisofs -o home.iso -J -R \                Присоединяет файлы к ветви с образом
 -graft-points Pictures/=/usr/share/pixmaps/ \
 /home/chris
```

В первом примере показаны различные файлы и каталоги, объединенные и помещенные в корень образа ISO. Во втором примере содержимое каталога /var/pics присоединяется к каталогу /home/chris/Pictures, в результате чего на компакт-диске с образом в каталоге /Pictures будет находиться все содержимое каталога /usr/share/pixmaps.

Добавление информации в заголовок образа ISO позже может определить содержимое этого образа, что особенно полезно, если образ сохраняется или распространяется через сеть, то есть без использования физического диска. Вот несколько примеров:

```
$ mkisofs -o /tmp/home.iso -R -J \      Добавляет информацию в заголовок ISO
   -p www.handsonhistory.com \
   -publisher "Swan Bay Folk Art Center" \
   -V "WebBackup" \
   -A "mkisofs" \
   -volset "1 of 4 backups, July 30, 2007" \
   /home/chris
```

В данном примере параметр -p обозначает ID создателя образа ISO, который может содержать номер телефона, почтовый адрес или адрес сайта, через который с ним можно связаться. С помощью параметра -publisher вы можете добавить до 128 символов информации о создателе образа (например, название компании или организации). Параметр -V указывает ID тома, который является важной информацией, поскольку на многих операционных системах Linux именно он используется для монтирования компакт-диска после его запуска. Например, на основании информации, содержащейся в приведенной выше командной строке, компакт-диск в Ubuntu и других системах Linux будет монтирован в каталог /media/WebBackup. Параметр -A может быть использован для обозначения приложения, с помощью которого создается образ ISO. Параметр -volset может содержать строку с информацией о настройке образов ISO.

После создания образа ISO и перед его записью на диск вы можете **проверить образ** и убедиться, что ко всем файлам, содержащимся на диске, может быть осуществлен доступ. Проверку можно произвести следующими способами:

```
$ volname home.iso          Отображает имя тома
WebBackup
$ isoinfo -d -i home.iso    Отображает информацию заголовка
CD-ROM is in ISO 9660 format
System id: LINUX
Volume id: WebBackup
Volume set id: All Website material on November 2, 2007
Publisher id: Swan Bay Folk Art Center
Data preparer id: www.handsonhistory.com
Application id: mkisofs
Copyright File id:
Abstract File id:
Bibliographic File id:
Volume set size is: 1
Volume set sequence number is: 1
Logical block size is: 2048
Volume size is: 23805
Joliet with UCS level 3 found
Rock Ridge signatures version 1 found
```

При создании файла образа вы можете отобразить и большее количество введенной в командную строку mkisofs информации. Если образ необходимо опубликовать, то определите на компакт-диске месторасположение файла, закрепляющего авторские права (параметр -copyright), абстрактного файла (параметр -abstract) и файла библиографии (параметр -biblio). Убедившись, что заголовок содержит всю необходимую информацию, попробуйте получить доступ к файлам ISO-образа, для чего смонтируйте его:

```
$ sudo mkdir /mnt/myimage                     Создает точку монтирования
$ sudo mount -o loop home.iso /mnt/myimage    Циклически монтирует ISO-образ
$ ls -l /mnt/myimage                          Проверяет содержимое ISO-образа
$ sudo umount /mnt/myimage                    Демонтирует образ после
                                              выполнения операции
```

Помимо выполнения проверки на наличие доступа к файлам и каталогам образа ISO, убедитесь, что подписи даты и времени, права собственности и доступа указаны верно. Позже эта информация может оказаться полезной, если понадобится восстановить прежнюю информацию.

апись образов

Команда cdrecord является наиболее популярным в среде Linux консольным приложением для записи образов на компакт-диски и DVD. После создания ISO-образа (как было описано ранее) или получения его другим способом (например, после загрузки установочного компакт-диска или Live CD из Интернета) cdrecord позволяет быстро записать этот образ на диск.

ПРИМЕЧАНИЕ

В Ubuntu вместо команды cdrecord используется команда wodim, созданная на базе кода cdrecord и поддерживающая те же параметры. Если вы попробуете выполнить команду cdrecord, то в этой версии Ubuntu, скорее всего, запустится wodim. Если у вас возникнут проблемы с этой утилитой, свяжитесь с проектом CDRkit (http://cdrkit.org).

Создание ISO-образов компакт-дисков и DVD выполняется абсолютно одинаково, за исключением того, что образ DVD может быть значительно больше, чем образ компакт-диска, поэтому проверьте объем имеющихся у вас в наличии носителей. Компакт-диски обычно имеют объем 650 Мбайт, 700 Мбайт или 800 Мбайт, тогда как мини-CD вмещают 50 Мбайт, 180 Мбайт, 185 Мбайт или 193 Мбайт. Однослойные DVD имеют объем 4,7 Гбайт, тогда как на двухслойные DVD помещается до 8,4 Гбайт.

ПРИМЕЧАНИЕ

Имейте в виду, что производители CD/DVD при расчете вместимости носителей размер одного мегабайта принимают равным 1000 Кбайт вместо 1024 Кбайт. Чтобы проверить, поместится ли ISO-образ на имеющийся у вас носитель, выполните команду du --si home.iso, позволяющую отобразить размер образа (вместо используемой обычно команды du -sh).

Перед началом записи на компакт-диск или DVD **убедитесь, что ваш привод поддерживает запись CD/DVD**, и укажите путь к носителю. Для этого воспользуйтесь параметром --scanbus команды cdrecord:

```
$ cdrecord -scanbus          Выводит список приводов, не поддерживающих
                             функцию записи
scsibus0:
     0,0,0 0) 'SAMSUNG ' 'DVD-ROM SD-616E ' 'F503' Removable CD-ROM
     0,0,0 1) *
     0,0,0 2) *
           ...
$ cdrecord -scanbus          Отображает список приводов, которые могут
                             записывать компакт-диски или DVD
scsibus0:
   0,0,0 0) 'LITE-ON ' 'DVDRW SOHW-1633S' 'BS0C' Removable CD-ROM
   0,0,0 1) *
   0,0,0 2) *
         ...
```

Первый из приведенных примеров отображает привод CD/DVD, поддерживающий только чтение и не записывающий диски (DVD-ROM и CD-ROM). Команда, приведенная во втором примере, отображает привод, который способен осуществлять запись на компакт-диски или DVD (DVD-RW). Вставьте носитель, на который хотите произвести запись, в привод. Ниже приведены примеры команд cdrecord, позволяющих **записывать образы на компакт-диски или DVD**:

```
$ cdrecord -dummy home.iso      Проверяет привод перед записью
$ cdrecord -v home.iso          Выполняет запись образа на компакт-диск,
                                отображая детальную информацию
                                (настройки по умолчанию)
```

```
$ cdrecord -v speed=24 home.iso      Устанавливает скорость записи
$ cdrecord -pad home.iso             При невозможности чтения дорожки
                                     добавляет 15 нулевых секторов
$ cdrecord -eject home.iso           Извлекает CD/DVD после завершения
                                     записи
$ cdrecord /dev/cdrw home.iso        Определяет привод по имени устройства
                                     (может отличаться)
$ cdrecord dev=0,2,0 home.iso        Определяет привод по имени SCSI
```

Команда cdrecord позволяет также **записывать мультисессионные CD/DVD**:

```
$ cdrecord -multi home.iso           Открывает мультисессию
$ cdrecord -msinfo                   Проверяет номер сессии для следующего прожига
Using /dev/cdrom of unknown capabilities
0,93041
$ mkisofs -J -R -o new.iso \         Создает еще один ISO-образ для записи
  -C 0,93041 /home/chris/more        Задает начальную точку и определяет новые
                                     данные для ISO
$ cdrecord new.iso                   Записывает новые данные на существующий
                                     компакт-диск
```

Вы можете использовать параметр -multi до тех пор, пока не заполните весь диск. Для последнего прожига, чтобы закрыть сессию записи компакт-диска, не используйте параметр -multi.

Создание и запись DVD

Используя команду growisofs, вы можете совмещать два этапа создания диска: группировку файлов в образ ISO (mkisofs) и запись этого образа на DVD (cdrecord). Помимо экономии времени, команда growisofs позволяет оставлять сессию записи открытой (по умолчанию) до тех пор, пока вы самостоятельно ее не закроете.

Ниже представлены примеры использования команд growisofs для работы с мультисессиями:

```
$ growisofs -Z /dev/dvd -R -J /home/chris       Подготавливает и выполняет
                                                запись образа на DVD
$ growisofs -Z /dev/dvd -R -J /home/francois    Добавляет файлы к образу
$ growisofs -M /dev/dvd=/dev/zero               Закрывает сессию
```

Если вы хотите воспользоваться дополнительными параметрами при записи образа ISO, то можете просто добавить в командную строку параметры команды mkisofs (например, обратите внимание на использование параметров -R и -J в предыдущих примерах).

Если вы хотите **записать** с помощью команды growisofs **образ DVD**, воспользуйтесь параметром -dvd-compat:

```
$ growisofs -dvd-compat -Z /dev/dvd=image.iso   Записывает образ ISO на DVD
```

Выполнение нескольких мультисессионных записей DVD с помощью параметра -dvd-compat может повысить совместимость с другими приводами DVD.

Резюме

В операционных системах Linux и предшествующих им системах UNIX резервное копирование данных осуществлялось посредством комбинирования команд, каждая из которых выполняла определенный набор функций. Резервное копирование важных данных и сегодня может выполняться таким же образом, однако сейчас уже существует множество приложений, выполняющих эти операции безопаснее и эффективнее.

Утилита для ленточного архивирования (команда `tar`), ранее используемая для резервного копирования данных на магнитную ленту, расширила свои основные функции. Поскольку практически во всех операционных системах Linux и UNIX присутствует утилита tar, то именно она стала стандартом программного обеспечения для упаковки и резервного копирования данных. Существуют различные способы передачи и хранения полученных архивов.

Для перемещения резервных копий данных на другие компьютеры сети вы можете использовать службы удаленного исполнения OpenSSH (такие как ssh), а также приложение rsync, с помощью которого вы можете экономить системные ресурсы, копируя только изменившиеся файлы (или их части).

Невысокая стоимость компакт-дисков и DVD сделала эти носители популярными для резервного копирования личных данных или данных небольшого офиса. Команда `mkisofs` может создавать из скопированных резервных данных файловые системы в формате ISO9660, воспринимаемом многими операционными системами (Linux, Windows или Mac). После создания с помощью команды `mkisofs` ISO-образа он может быть записан на компакт-диск или DVD посредством команды `cdrecord` или `growisofs`.

9 Проверка запущенных процессов и управление ими

В Linux любая запускаемая программа выполняется в виде процесса, отображаемого в таблице процессов Linux. Linux располагает всеми необходимыми приложениями для просмотра и изменения процессов, выполняющихся в системе.

Для просмотра информации о запущенных процессах наиболее удобны команды ps и top, для которых существуют десятки параметров, позволяющих отображать необходимую информацию о процессах.

Кроме того, в Linux существует команда pgrep, которая может помочь в поиске необходимого процесса.

Существуют также такие команды, как nice и renice, используемые для повышения или уменьшения уровня использования ресурсов процессора тем или иным процессом. Вместе с тем вы можете изменять статус процесса на фоновый (команда bg) и приоритетный (команда fg).

Одним из способов изменения выполняемых процессами действий или полного их завершения является отправление процессам определенных сигналов. Используя команды kill и killall, вы можете отправлять процессам сигналы по их PID или именам соответственно. Вы можете отправлять процессам и другие сигналы, заставляющие их выполнять, например повторную проверку конфигурационных файлов или продолжать выполнение ранее остановленного процесса.

Для составления расписания выполнения команд или работы с процессами, которые не являются частью вашей текущей сессии работы с консолью, вы можете использовать команды at и batch. Для запуска команд в установленное время предназначены утилиты cron и anacron. Кроме того, вы можете перемещать тексты колов (или символьные ссылки на них) в каталог /etc/cron.hourly (а также cron.daily, cron.weekly или cron.monthly).

Отображение активных процессов

Для отображения списка запущенных в данный момент в системе процессов, как правило, используются команды ps и top. Команда ps выводит информацию (в виде обычного списка) о запущенных в данный момент процессах. Команда же top ориентирована на работу с экраном и отображает постоянно обновляющийся

список запущенных команд, сортируя его по указанным критериям (уровень использования времени центрального процессора, оперативной памяти, UID и т. д.).

Команда ps

В каждой операционной системе Linux (а также во всех системах, производных от UNIX, таких как BSD, Mac OS X и др.) присутствует команда ps, однако с годами появилось множество версий данной команды, незначительно отличающихся от начальной версии некоторыми параметрами. Поскольку команда ps появилась вместе с первыми системами UNIX, в ней используется нестандартный ввод параметров (например, в некоторых случаях вам потребуется использовать перед параметром знак дефиса).

Различные примеры команд, приведенные в данной главе и демонстрирующие способы использования команды ps, применимы как к Ubuntu, так и к большинству других систем Linux. Ниже представлено несколько примеров, которые вы можете использовать для **отображения процессов текущего пользователя** (в табл. 9.1 приведено описание отображаемых командой ps столбцов):

```
$ ps            Отображает в консоли процессы текущего пользователя
PID     TTY      TIME       CMD
2552    pts/0    00:00:00   bash
3438    pts/0    00:00:00   ps
$ ps -u chris   Отображает все процессы, запущенные пользователем
                chris (стандартный вывод)
PID       TTY    TIME    COMMAND
2678      tty1   0:00    startx
2689      tty1   0:00    xinit
2710      tty1   0:06    gnome-session
 ...
$ ps -u chris u        Отображает все процессы, запущенные пользователем
                       chris (с учетом использования CPU/MEM)
USER  PID  %CPU %MEM VSZ    RSS   TTY   STAT START TIME COMMAND
chris 2678 0.0  0.0  4328   852   tty1  S+   Aug14 0:00 /bin/sh startx
chris 2689 0.0  0.1  2408   488   tty1  S+   Aug14 0:00 xinit
chris 2710 0.0  1.1  22016  496   tty1  S    Aug14 0:06 gnome-session
 ...
$ ps -fu chris              Отображает все процессы, запущенные пользователем
                            chris (с учетом PPID)
UID    PID   PPID  C STIME TTY  TIME        CMD
chris  2678  2645  0 Aug14 tty1 00:00:00    /bin/sh /usr/X11R6/bin/startx
chris  2689  2678  0 Aug14 tty1 00:00:00    xinit /etc/X11/xinit/xinitrc
chris  2710  2689  0 Aug14 tty1 00:00:09    /usr/bin/gnome-session
 ...
$ ps -Fu chris              Отображает все процессы, запущенные пользователем
                            chris (с учетом SZ и PSR)
UID    PID   PPID  C  SZ  RSS PSR STIME  TTY  TIME      CMD
chris  2678  2645  0 1082  852  0  Aug14 tty1 00:00:00  /bin/sh startx
chris  2689  2678  0  602  488  0  Aug14 tty1 00:00:00  xinit
chris  2710  2689 0 5504 5440  0  Aug14 tty1 00:00:09 gnome-session
```

В данных примерах отображаются некоторые процессы, запущенные в сеансе оконного менеджера GNOME. В первом примере выполняется команда ps, запущенная из окна терминала, поэтому в данном окне вы видите только процессы текущей консоли. В других примерах демонстрируются способы отображения различной информацию о процессах (примеры, демонстрирующие порядок формирования выходной информации команды на основе индивидуально заданных параметров, будут приведены позже). Описания столбцов ps приведены в табл. 9.1.

Ниже показаны примеры использования команды ps для **отображения каждого выполняемого в системе процесса**:

```
$ ps -e        Отображает каждый выполняемый процесс
PID TTY        TIME           CMD
  1   ?        00:00:01    init
  2   ?        00:00:00    migration/0
  3   ?        00:00:00    ksoftirqd/0
 ...
$ ps -el       Отображает более подробную информацию о каждом
               выполняемом процессе
F S   UID   PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY   TIME       CMD
4 S     0     1     0  0  75   0 -   534 -       ?    00:00:01   init
1 S     0     2     1  0 -40   - -     0 -       ?    00:00:00   migration/0
1 S     0     3     1  0  94  19 -     0 -       ?    00:00:00   ksoftirqd/0
  ...
$ ps -ef       Выводит полную информацию о каждом
               выполняемом процессе
UID     PID   PPID  C   STIME  TTY    TIME       CMD
root    1     0     0   Aug05  ?      00:00:01   init [5]
root    2     1     0   Aug05  ?      00:00:00   [migration/0]
root    3     1     0   Aug05  ?      00:00:00   [ksoftirqd/0]
  ...
$ ps -eF       Отображает расширенную информацию о каждом
               выполняемом процессе
UID   PID  PPID  C  SZ   RSS   PSR  STIME  TTY TIME      CMD
root  1    0     0  534  556     0  Aug05  ?   00:00:01  init [5]
root  2    1     0    0    0     0  Aug05  ?   00:00:00  [migration/0]
root  3    1     0    0    0     0  Aug05  ?   00:00:00  [ksoftirqd/0]
  ...
$ ps ax        Выводит информацию о каждом выполняемом
               процессе в сжатом стиле BSD
PID TTY        STAT        TIME COMMAND
1   ?          Ss          0:01 init [5]
2   ?          S           0:00 [migration/0]
3   ?          SN          0:00 [ksoftirqd/0]
  ...
$ ps aux       Отображает информацию о каждом выполняемом
               процессе в полном стиле BSD
USER    PID   %CPU %MEM  VSZ   RSS TTYSTAT START  TIME COMMAND
root    1     0.0  0.0   2136  556 ?        Ss    Aug05  0:01 init [5]
root    2     0.0  0.0      0    0 ?        S     Aug05  0:00 [migration/0]
root    3     0.0  0.0      0    0 ?        SN    Aug05  0:00 [ksoftirqd/0]
```

```
$ ps auwx              Отображает информацию о каждом выполняемом
                       процессе в полном стиле BSD и расширенном формате
$ ps auwwx             Выводит информацию о каждом выполняемом
                       процессе в полном стиле BSD и неограниченной шириной
```

Некоторые процессы могут запускать собственные процессы. Например, веб-сервер создает множество процессов httpd, которые ожидают запросов на сервере. Используя различные параметры команды ps, вы можете **отображать процессы в иерархическом порядке (в виде дерева)**:

```
$ ps -ejH              Отображает процессы в иерархическом порядке
                       на основании ID процессов или сессий
  PID  PGID   SID    TTY      TIME       CMD
    1     1     1    ?        00:00:01   init
    2     1     1    ?        00:00:00   migration/0
 2043  2043  2043    ?        00:00:00   sshd
 2549  2549  2549    ?        00:00:00   sshd
 2551  2549  2549    ?        00:00:00   sshd
 2552  2552  2552    pts/0    00:00:00   bash
 7760  7760  7760    ?        00:00:00   httpd
 7762  7760  7760    ?        00:00:00   httpd
 7763  7760  7760    ?        00:00:00   httpd
$ ps axjf      Отображает процессы в иерархическом порядке в стиле BSD
PPID   PID  PGID   SID TTY      TPGID STAT  UID   TIME COMMAND
    0     1     1     1 ?          -1 Ss      0   0:01 init [5]
    1     2     1     1 ?          -1 S       0   0:00 [migration/0]
    1  2043  2043  2043 ?          -1 Ss      0   0:00 /usr/sbin/sshd
 2043  2549  2549  2549 ?          -1 Ss      0   0:00  \_ sshd: chris [priv]
 2549  2551  2549  2549 ?          -1 S     500   0:00  | \_ sshd: chris@pts
 2551  2552  2552  2552 pts/0    8398 Ss    500   0:00  | \_ -bash
    1  7760  7760  7760 ?          -1 Ss      0   0:00 /usr/sbin/httpd
 7760  7762  7760  7760 ?          -1 S      48   0:00  \_ /usr/sbin/httpd
 7760  7763  7760  7760 ?          -1 S      48   0:00  \_ /usr/sbin/httpd
$ ps -ef -forest       Отображает процессы в иерархическом порядке в виде леса
UID        PID  PPID  C STIME TTY        TIME     CMD
root         1     0  0 Aug05 ?          00:00:01 init [5]
root         2     1  0 Aug05 ?          00:00:00 [migration/0]
root         3     1  0 Aug05 ?          00:00:00 [ksoftirqd/0]
root      2043     1  0 Aug05 ?          00:00:00 /usr/sbin/sshd
root      2549  2043  0 Aug16 ?          00:00:00  \_ sshd: chris [priv]
chris     2551  2549  0 Aug16 ?          00:00:00  | \_ sshd: chris@pts/0
chris     2552  2551  0 Aug16 pts/0      00:00:00  | \_ -bash
root      7760     1  0 18:27 ?          00:00:00 /usr/sbin/httpd
apache    7762  7760  0 18:27 ?          00:00:00  \_ /usr/sbin/httpd
apache    7763  7760  0 18:27 ?          00:00:00  \_ /usr/sbin/httpd
$ pstree Отображает процессы в алфавитном порядке в виде дерева
init-+-Xorg
     |-at-spi-registry
     |-atd
     |-auditd-+-audispd
     |        `-{auditd}
```

```
 |-sshd-+-sshd---sshd---bash---pstree
 |      |-sshd---sshd---bash---su---bash
 |      `-sshd---sshd---bash---su---bash---su---bash---vim
...
```

Показанные примеры демонстрируют различные способы отображения процессов в иерархическом порядке. Чтобы сравнить некоторые одинаковые процессы, отображающие различную информацию, текст выхода был сокращен. Стоит отметить, что PPID (ID порождающего процесса) является ID процесса, запустившего соответствующий дочерний процесс. Процессы sshd отображают запущенную ssd с вошедшим в систему через сеть пользователем и выполняют запуск консоли bash (и в конечном счете редактор vim). Демон httpd обеспечивает работоспособность веб-сервера Apache, в котором порождающий процесс запускается суперпользователем, а дочерние процессы — пользователем. В последнем примере приведена команда pstree, которая обычно используется для отображения дерева процессов.

Если вы предпочитаете самостоятельно определять набор отображаемых командой ps данных, то воспользуйтесь параметром -o. После этого для сортировки отображенной информации по любым данным вы сможете использовать параметр --sort. В табл. 9.1 приведен перечень возможных отображаемых командой ps данных и соответствующие им параметры, которые необходимо использовать вместе с параметром -o, чтобы отобразить необходимый столбец.

Таблица 9.1. Отображаемые командой ps данные и соответствующие им параметры

Параметр	Заголовок столбца	Описание
%cpu	%CPU	Уровень использования времени центрального процессора за время существования процесса в формате 00.0
%mem	use (resident set size)	Уровень использования процессом физической памяти (в процентах)
args	COMMAND	Команда со всеми аргументами
bsdstart	START	Время запуска команды в формате ЧЧ:ММ или Месяц:День
bsdtime	TIME	Полное (включая пользователя и систему) время использования времени центрального процессора
comm	COMMAND	Только имя команды (без аргументов)
cp	CP	Уровень использования времени центрального процессора в десятых долях процента
cputime	TIME	Полное время использования центрального процессора в формате [DD-]HH:MM:SS
egid	EGID	Эффективный идентификатор группы процесса (целое число)
egroup	EGROUP	Эффективный идентификатор группы процессов (имя)
etime	ELAPSED	Время запуска процесса в формате [[DD-]HH:]MM:SS
euid	EUID	Эффективный идентификатор пользователя, запустившего процесс (одно целое)
euser	EUSER	Эффективный идентификатор пользователя, запустившего процесс (имя)

Продолжение ⇗

Таблица 9.1 (продолжение)

Параметр	Заголовок столбца	Описание
fgid	FGID	Идентификатор группы доступа к файловой системе (номер)
fgroup	FGROUP	Идентификатор группы доступа к файловой системе (имя)
fname	COMMAND	Первые восемь символов имени команды
fuid	FUID	Идентификатор пользователя, имеющего доступ к файловой системе (номер)
fuser	FUSER	Идентификатор пользователя, имеющего доступ к файловой системе (имя)
lstart	STARTED	Дата и время запуска команды
nice	NI	Уровень загрузки процессора (от 19 (наименьшая) до –20 (максимальная загрузка))
pgid	PGID	Идентификатор группы процесса
pid	PID	Идентификационный номер процесса
ppid	PPID	Идентификатор родительского процесса
psr	PSR	Принадлежность к процессам процессора (первый центральный процессор соответствует нулю)
rgid	RGID	Текущий идентификатор группы (номер)
rgroup	RGROUP	Текущая группа (имя)
rss	RSS	Объем используемой физической памяти (без учета области подкачки) в Кбайт
rtprio	RTPRIO	Приоритет реального времени
ruid	RUID	Идентификатор текущего пользователя (номер)
ruser	RUSER	Текущий пользователь (имя)
s	S	Односимвольный индикатор состояния (D — спящий, не может быть прерван; R — выполняющийся; S — спящий, может быть прерван; T — остановленный; W — «слушает»; X — неактивный; Z — «зомби»)
sess	SESS	Идентификатор заголовка сессии
sgi_p	P	Текущий используемый процессор
size	SZ	Приблизительный объем области подкачки, необходимый для того, чтобы процесс был выгружен
start	STARTED	Время запуска команды в формате ЧЧ:ММ:СС или Месяц День
start_time	START	Время запуска команды: ЧЧ:ММ или Месяц День
stat	STAT	Расширенный индикатор состояния: буква «s» совместно с другими символами (< — высокий приоритет; N — низкий приоритет; L — страницы зафиксированы в памяти; s — определяет, является ли сессия главной; l — многопоточный; + — находится в группе фоновых процессов)
sz	SZ	Размер образа ядра процесса (физические страницы)
tname	TTY	Контролирующий tty (терминал)
user	USER	Идентификатор действительного пользователя (имя)
vsize	VSZ	Объем виртуальной памяти процесса (1024-битные единицы)

Отметим, что некоторые значения, предназначенные для отображения имен пользователей, могут выводить номера (UID), если имя является слишком длинным, чтобы поместиться в имеющемся пространстве.

С помощью параметров столбцов, разделенных запятыми, вы можете сформировать собственный **индивидуальный формат отображения данных**. Ниже представлено несколько примеров подобного отображения данных текущих процессов:

```
$ ps -eo ppid,user,%mem,size,vsize,comm --sort=-size Сортирует процессы
                                                     по уровню использования
                                                     оперативной памяти
PPID USER     %MEM    SZ    VSZ COMMAND
   1 root     27.0 68176  84264 yum-updatesd
$ ps -eo ppid,user,bsdstart,bsdtime,%cpu,args --sort=-%cpu Сортирует процессы
                                                     по уровню использования
                                                     процессорного времени
PPID USER      START   TIME %CPU COMMAND
   1 root     Jul 30  44:20 27.1 /usr/bin/python /usr/sbin/yum-updatesd
$ ps -eo ppid,user,nice,cputime,args --sort=-nice   Сортирует процессы
                                                     по приоритету, от меньшего
                                                     к большему
PPID USER     NI     TIME COMMAND
   1 root     19 00:44:26 /usr/bin/python /usr/sbin/yum-updatesd
$ ps -eo ppid,user,stat,tname,sess,cputime,args --sort=user Сортирует
                                                     процессы по имени
                                                     пользователя
PPID USER     STAT TTY    SESS     TIME COMMAND
   1 avahi     Ss ?       2221 00:00:07 avahi-daemon: running [example.net]
```

Вот еще несколько **примеров использования команды** ps:

```
$ ps -C httpd   Отображает запущенные httpd-процессы
  PID TTY          TIME CMD
 1493 ?        00:00:00 httpd
 1495 ?        00:00:00 httpd
```

Отметим, что для запуска процесса httpd необходимо установить сервер HTTP, например Apache.

```
$ ps -p 5413 -o pid,ppid,bsdtime,args               Отображает информацию
                                                     о процессе PID 5413
  PID  PPID TIME COMMAND
 5413     1 0:08 gpm -m /dev/input/mice -t exps2
$ ps -U chris,francois -o pid,ruser,tty,stat,args   Отображает информацию
                                                     о двух пользователях
  PID RUSER    TT       STAT COMMAND
 1010 chris    pts/0    Ss   -bash
 5951 francois pts/1    Ss+  /bin/bash
```

Команда top

Если вам необходимо **непрерывно просматривать запущенные в системе процессы**, то воспользуйтесь командой top. Эта команда ориентирована на работу

с экраном и отображает постоянно обновляющийся список запущенных процессов. Если вы выполните команду top без параметров, то она отобразит время работы системы, список выполняемых задач, уровень потребления оперативной памяти и уровень использования процессорного времени, по которому отсортирует полученный список:

```
$ top
top - 01:39:43 up 4 days, 1:53, 6 users, load average: 1.25, 1.08, 1.11
Tasks: 119 total, 1 running, 117 sleeping, 0 stopped, 1 zombie
Cpu(s): 46.8% us,  3.3% sy,  0.0% ni,  49.5% id,  0.0% wa, 0.3% hi, 0.0% si
Mem:    482992k total,   472688k used,    10304k free,   24312k buffers
Swap:  5863716k total,   534512k used,  5329204k free,   68072k cached

  PID USER      PR  NI  VIRT  RES  SHR  S %CPU %MEM  TIME+   COMMAND
 2690 root      15   0  344m  76m 7116  S 32.2 16.2  2349:08 X
 2778 chris     15   0 16212 7992 4836  S  1.7  1.7  4:30.61 metacity
22279 chris     15   0  227m 109m  23m  S  1.0 23.3 34:34.00 firefox-bin
```

Ниже приведены еще некоторые примеры параметров, которые вы можете использовать для **отображения непрерывно обновляющегося списка запущенных процессов**:

```
$ top -d 5             Устанавливает время обновления равным 5 секундам
                       (по умолчанию 3)
$ top -u francois      Отображает только процессы действительного
                       пользователя francois
$ top -p 190,2690      Отображает только процессы 190 и 2690
$ top -n 10            После десяти обновлений завершает выполнение команды
$ top -b               Запускается в режиме, не рассчитанном на работу с экраном
```

В последнем примере (top -b) выход команды форматируется в подходящей для выхода файла форме, противоположной формирующейся при повторном выводе того же экрана для интерактивного просмотра. Данная команда может использоваться для создания журнала процессов, например для отслеживания процессов, потребляющих все ресурсы посреди ночи. С помощью следующей команды можно **запустить команду top, фиксирующую информацию о процессах** в течение десяти часов:

```
$ top -b -n 12000 > myprocesslog
```

Во время выполнения команды top вы можете **обновлять и различными способами сортировать список процессов**. Чтобы **немедленно обновить список процессов**, нажмите клавишу Пробел или Enter. Для сортировки процессов **по их PID** нажмите сочетание клавиш Shift+N, для сортировки процессов **по потреблению процессорного времени** — Shift+P, для сортировки **по потреблению оперативной памяти** — Shift+M, а для сортировки **по времени потребления процессора** — Shift+T. Чтобы изменить столбец для сортировки, используйте символы < (**упорядочить столбец слева**) и > (**упорядочить столбец справа**) или, **указав букву столбца, по которой хотите отсортировать список**, нажмите клавишу f, когда отобразится список столбцов.

Во время выполнения команды top вы можете **различными путями изменять действия команды**. Чтобы **изменить период обновления данных**, нажмите клави-

шу d и введите число, обозначающее количество секунд. Для **отображения процессов только конкретного пользователя** нажмите клавишу u и введите имя пользователя. Чтобы **отобразить заданное количество процессов**, нажмите клавишу n и укажите число, обозначающее требуемое количество процессов. **Для возврата в начальный экран** программы нажмите =.

Кроме того, с помощью программы top вы можете **влиять на многие запущенные процессы**. Чтобы **отправить работающему процессу сигнал о его завершении**, нажмите клавишу K и затем укажите PID процесса, которому хотите отправить сигнал. Затем для отправления другого сигнала укажите другое число или нажмите клавишу 9 для завершения действия. Чтобы **увеличить или уменьшить приоритет процесса**, нажмите клавишу N, а затем отрицательное число (для повышения приоритета) или положительное число (для уменьшения приоритета)

Чтобы **получить более подробную информацию** об использовании команды top, в ходе ее выполнения нажмите ?. MAN-страница также содержит много полезной информации:

```
$ man top        Отображает MAN-страницу, посвященную использованию команды top
```

Для завершения выполнения команды top нажмите клавишу Q.

Поиск процессов и управление ими

Изменение текущего процесса прежде всего означает поиск соответствующего процесса и лишь затем уже изменение его приоритета или подачу процессу сигнала об изменении режима работы. Если вы ищете какой-то определенный процесс, то наиболее простым вариантом будет отобразить расширенный список процессов с помощью команд ps или top. Команда pgrep позволяет искать необходимые процессы через активные процессы, команда renice — изменять приоритеты текущих процессов, а команды kill, pkill и killall — отправлять сигналы текущим процессам (включая сигналы завершения данных процессов).

Команда pgrep

С помощью простого использования команды pgrep вы можете осуществлять поиск по имени команды (или части имени) и отображать ID всех процессов, содержащих данное имя:

```
$ pgrep init    Отображает ID всех процессов, содержащих «init»
1
2689
```

Поскольку мы знаем, что существует только одна текущая команда init, то в дальнейшем, чтобы отобразить все имена команд процессов (а также узнать, почему обнаружено два процесса), будет использоваться параметр -l:

```
$ pgrep -l init        Отображает PID и имена всех процессов, содержащих «init»
1 init
2689 xinit
```

Кроме того, вы можете **искать процессы, принадлежащие конкретному пользователю**:

```
$ pgrep -lu chris     Перечисляет все процессы, принадлежащие пользователю chris
2551 sshd
2552 bash
2803 vim
```

Возможно, самым эффективным способом использования команды pgrep является **поиск ID текущих процессов и передача PID другим командам** для формирования отображаемой информации. Вот несколько примеров (если у вас процессы metacity и firefox не запущены, укажите другие):

```
$ ps -p `pgrep metacity`    Осуществляет поиск процесса metacity и выполняет
                            команду ps (краткий вывод)
  PID TTY          TIME CMD
 2778 ?        00:05:00 metacity
$ ps -fp $(pgrep nautilus)  Осуществляет поиск процесса nautilus и выполняет
                            команду ps (полный вывод)
UID        PID  PPID  C STIME TTY          TIME CMD
chris     5907  5840  0 Sep05 ?        00:00:26 nautilus --no-default-window --s
# sudo# renice -5 $(pgrep firefox)       Выполняет поиск процесса firefox
                                         и повышает приоритет команды
20522: old priority 0, new priority -5
20557: old priority 0, new priority -5
```

Команды, принимающие ID процессов в качестве входящей информации, в данном случае могут объединяться с командой pgrep. Как видно из предыдущего примера использования команды pgrep, вы можете применять такие команды, как renice, чтобы изменять режим работы текущего процесса.

Команда fuser

Другим способом выполнения поиска отдельного процесса является поиск по элементу, осуществляющему доступ к процессу. Команда fuser может использоваться для определения файлов или сокетов, в которых выполняются какие-либо процессы. После обнаружения процесса fuser может использоваться для отправления сигналов этим процессам.

Команда fuser наиболее полезна для поиска файлов смонтированных файловых систем, используемых процессами (как на локальных жестких дисках или общих ресурсов Samba). Обнаружение этих процессов позволяет правильно закрывать их (или при необходимости уничтожать) и затем безопасно демонтировать файловую систему.

Вот несколько примеров применения команды fuser для **отображения процессов, использующих файлы в данной файловой системе**:

```
$ fuser -mauv /boot   Отображает подробный список процессов, открывших /boot
                      USER      PID ACCESS COMMAND
```

```
/boot/grub/:    root    3853 ..c..  (root)bash
                root   19760 ..c..  (root)bash
                root   28171 F.c..  (root)vi
                root   29252 ..c..  (root)man
                root   29255 ..c..  (root)sh
                root   29396 F.c..  (root)vi
```

В приведенном примере отображаются идентификаторы запущенных процессов, ассоциированных с файловой системой /boot. Этим процессам должен соответствовать открытый файл, консоль, или они должны являться дочерними процессами консоли с рабочим каталогом, расположенным в /boot. В данном примере в файловой системе /boot присутствуют две открытые консоли bash, а также выполняются две команды vi с открытыми в этой файловой системе файлами и команда man. Параметр -a позволяет отобразить полный список процессов, -u отображает имя пользователя или процесса, запустившего данный процесс, а -v формирует подробный выход.

Ниже приведены другие примеры использования команды fuser для **отображения процессов с открытыми файлами**:

```
$ fuser /boot          Отображает родительские PID процессов,
                       использующих файлы в /boot
/boot:        19760c 29396c
$ fuser -m /boot       Отображает все PID процессов, использующих файлы в /boot
/boot:        3853c 19760c 28171c 29396c 29252c 29255c
$ fuser -u /boot       Отображает PID или имя пользователя консоли,
                       работающей в /boot
/boot:        19760c(root) 29396c(root) 29252c(root) 29255c(root)
```

После определения процессов, открывших файлы, вы сможете приостановить их вручную или уничтожить (полностью завершить). По возможности всегда вручную останавливайте процессы, поскольку при уничтожении после них могут оставаться нежелательные файлы. Ниже приведены примеры использования команды fuser для **уничтожения, а также отправления других сигналов всем процессам, файлы которых используются в файловой системе**:

```
$ sudo fuser -k /boot       Уничтожает все процессы с открытыми файлами
                            в /boot (SIGKILL)
$ fuser -l                  Отображает список поддерживаемых сигналов
HUP INT QUIT ILL TRAP ABRT IOT BUS FPE KILL USR1 SEGV USR2 PIPE ALRM TERM STKFLT
CHLD CONT STOP TSTP TTIN TTOU URG XCPU XFSZ VTALRM PROF WINCH IO PWR SYS UNUSED
$ sudo fuser -k -HUP /boot  Отправляет HUP-сигнал всем процессам,
                            работающим с файловой системой /boot
```

Изменение запущенных процессов

Вы можете различными способами изменять параметры работы процесса даже после его запуска. Используя команду renice, описанную ранее, вы можете в планировщике своей операционной системы определять приоритет запущенного

процесса. Используя же команду nice, вы можете установить приоритет по умолчанию, а также повысить или понизить приоритет процесса во время его запуска.

Другим способом управления запущенными процессами является отправление этим процессам сигналов. Команды kill и killall могут использоваться для отправления процессам сигналов завершения. Кроме того, для этих целей может использоваться команда pkill.

Установка приоритета процессора с помощью команды nice

Каждый запущенный процесс поддерживает возможность управления значением параметра nice, который может использоваться для указания планировщику процессов Linux уровня приоритета данного процесса. Положительные значения параметра nice обычно понижают приоритет процесса. Идея использования этого параметра берет свое начало в объемных многопользовательских системах UNIX, в которых, чтобы дать возможность другим пользователям использовать процессор, необходимо было понизить приоритет основного процесса, «поступить правильно» («nice»).

Параметр nice не оказывает влияния на приоритет планировщика, а является лишь вариантом для него. Чтобы **определить текущее значение параметра nice**, выполните команду nice без параметров:

```
$ nice          Определяет текущее значение nice
0
```

Значение nice по умолчанию равно 0, однако, используя команду nice, вы можете уменьшать или увеличивать это значение. Значение приоритета может варьироваться от -20 (наивысший приоритет планировщика) до 19 (низший приоритет планировщика). Суперпользователь может как уменьшать, так и увеличивать значение nice любого пользователя, тогда как обычный пользователь может только уменьшать приоритет выполнения процессов (устанавливая большее значение nice).

ВНИМАНИЕ

Устанавливайте отрицательные значения приоритета выполнения процесса с особой осторожностью. Снижение приоритета системных процессов может привести к потере работоспособности системы.

Вот несколько примеров выполнения команды nice для **изменения значения nice определенной команды**:

```
$ nice -n 12 nroff -man a.roff | less   Выполняет форматирование MAN-страниц
                                        при пониженном приоритете
$ sudo nice -n -10 gimp                 Повышает приоритет команды gimp
```

С помощью команды renice вы можете изменять значение параметра nice уже после запуска процесса:

```
$ renice +2 -u francois                 Увеличивает значение nice процессов
                                        пользователя francois на 2
```

```
$ renice +5 4737                         Увеличивает значение nice процессов
                                         пользователя PID 4737 на 5
$ sudo renice -3 `pgrep -u chris spamd`  Понижает значение процессов spamd
                                         пользователя chris на 3
9688: old priority -1, new priority -3
20279: old priority -1, new priority -3
20282: old priority -1, new priority -3
```

В предыдущей командной строке для обозначения того, что выход команды pgrep отправляется команде renice (предполагается, что PID процессов spamd запущены пользователем **chris**), используются косые кавычки.

При выполнении команды top значения nice процессов отображаются по умолчанию. Кроме того, просмотреть настройки nice можно с помощью параметра -o nice при индивидуальном формировании выхода команды ps.

Запуск процессов в фоновом и приоритетном режимах

По умолчанию процесс из консоли запускается в приоритетном режиме. Это означает, что вы не можете выполнить другую команду до завершения данного процесса. Однако, добавив знак & в конец командной строки, вы можете запустить командную строку в фоновом режиме, а с помощью команд fg, bg и jobs вы можете переключать команды из фонового режима в приоритетный на всем протяжении всего времени выполнения управляющих командных строк.

Посредством следующей последовательности команд можно из окна терминала запустить программу для просмотра изображений GIMP, после чего с помощью контрольных клавиш **осуществлять запуск и остановку процессов и переключаться между фоновым и приоритетным режимами**:

```
$ gimp          Запускает gimp в фоновом режиме
<Ctrl+Z>        Останавливает процессы и перемещает их в фоновый режим
[1]+  Stopped   gimp
$ bg 1          Заново в фоновом режиме запускает работающие процессы
$ fg 1          Продолжает выполнение процессов в фоновом режиме
gimp
<Ctrl+C> Уничтожает процесс
```

Стоит отметить, что процессам, работающим в фоновом режиме, присваивается рабочий идентификационный номер (в данном случае 1). Поместив перед номером знак процента (например, %1) или просто указав номер вместе с командой (fg 1), вы можете определить конкретные фоновые процессы для команд bg и fg. При наличии одного или более процессов, запущенных в текущей консоли, вы можете использовать **для управления фоновыми процессами** команду jobs:

```
$ jobs          Отображает список процессов текущей консоли,
                выполняющихся в фоновом режиме
[1]    Running        gimp &
[2]    Running        xmms &
[3]-   Running        gedit &
[4]+   Stopped        gtali
$ jobs -l       Отображает PID вместе с информацией о каждом процессе
```

```
[1]  31676 Running          gimp &
[2]  31677 Running          xmms &
[3]- 31683 Running          gedit &
[4]+ 31688 Stopped          gtali
$ jobs -l %2                Отображает информацию только для процесса под номером %2
 [2]  31677 Running         xmms &
```

Приведенные выше примеры команд (jobs) могут использоваться, если необходимо во время входа в удаленную систему (используя ssh) запустить удаленные приложения GUI на локальном компьютере. Запустив эти процессы в фоновом режиме, вы можете работать сразу в нескольких приложениях, сохраняя их связь с текущей консолью. С помощью команды disown вы можете **отделить приложение от консоли во время его выполнения**:

```
$ disown %3          Отделяет процесс %3 от текущей консоли
$ disown -a          Отделяет все процессы от текущей консоли
$ disown -h          Ограничивает все процессы от действия HUP-сигнала,
                     отправленного в текущую консоль
```

Отделив процессы от консоли с помощью команды disown, вы можете закрывать консоль без необходимости уничтожать выполняемые процессы.

ПРИМЕЧАНИЕ

Если при использовании команд fg, bg или disown вы не укажете ID процесса, то будет задействован текущий процесс, обозначенный знаком +, следующим сразу за его названием.

Команды fg и bg позволяют управлять запущенными процессами, делая их фоновыми или приоритетными. Другим способом управления командами является отправление сигналов процессам напрямую. Наиболее распространенным способом отправления сигналов запущенным процессам является использование команд kill и killall.

Уничтожение процессов и отправление им сигналов

Отправляя запущенным процессам сигналы, вы можете останавливать их или вносить в них изменения. Такие команды, как kill и killall, могут отправлять процессам определяемые вами сигналы, заключающие в себе имена процессов. Чаще всего данные команды используются для уничтожения процессов.

Каждому сигналу соответствуют определенное число (9, 15 и т. д.) и строка (SIGKILL, SIGTERM и т. д.). В табл. 9.2 приведен перечень стандартных сигналов, которые могут отправляться процессам Linux.

Таблица 9.2. Стандартные сигналы, отправляемые процессам

Номер сигнала	Имя сигнала	Описание
1	SIGHUP	Сигнал завершения процесса терминала или контрольного процесса
2	SIGINT	Отправка с клавиатуры сигнала прерывания

Номер сигнала	Имя сигнала	Описание
3	SIGQUIT	Отправка с клавиатуры сигнала прекращения
4	SIGILL	Неразрешенная команда
6	SIGABRT	Запрет на выполнение, отправляемый функцией abort
8	SIGFPE	Исключение с плавающей точкой
9	SIGKILL	Сигнал полного завершения (уничтожения) процесса
11	SIGSEGV	Недействительная ссылка на ячейку памяти
13	SIGPIPE	Неработающий канал (нет процессов, которые могут быть считаны с канала)
14	SIGALRM	Сигнал таймера, обозначающий аварийный системный вызов
15	SIGTERM	Сигнал завершения
30, 10, 16	SIGUSR1	Определяемый пользователем сигнал 1
31, 12, 17	SIGUSR2	Определяемый пользователем сигнал 2
20, 17, 18	SIGCHLD	Завершенный или остановленный дочерний процесс
19, 18, 25	SIGCONT	Возобновление остановленного ранее процесса
17, 19, 23	SIGSTOP	Сигнал остановки процесса
18, 20, 24	SIGTSTP	Сигнал остановки, введенный в терминале
21, 21, 26	SIGTTIN	Ввод терминала для фоновых процессов
22, 22, 27	SIGTTOU	Вывод терминала для фоновых процессов

Команда `kill` может отправлять сигналы процессам в соответствии с их ID или номером задачи, тогда как команда `killall` может отправлять сигналы по имени процесса:

```
$ kill 28665                  Отправляет сигнал SIGTERM процессу PID 28665
$ kill -9 4895                Отправляет сигнал SIGKILL процессу PID 4895
$ kill -SIGCONT 5254          Возобновляет остановленный процесс (PID 5254)
$ kill %3                     Уничтожает процесс %3
$ killall spamd               Уничтожает все запущенные процессы spamd
$ killall -SIGHUP sendmail    Заставляет процесс sendmail перечитать
                              конфигурационные файлы
```

Сигнал `SIGKILL` (9), часто используемый неопытными администраторами-новичками, следует использовать в качестве последнего средства, поскольку он резко обрывает процесс, а не выполняет его аккуратное закрытие, что может привести к потере или повреждению данных, относящихся к данному процессу. Сигнал `SIGHUP` обычно использовался в системах UNIX для определения, отключен ли терминал от устройства (например, от модема hang-up или dial-in). Однако для перехвата сигналов `SIGHUP` с целью заставить процессы перечитать конфигурационные файлы были разработаны процессы, такие как, например, sendmail и httpd.

Отключение процессов от текущей консоли

Существует несколько способов закрыть консоль, не прерывая выполняемых процессов. Так, с помощью команды nohup можно **запустить процесс таким образом, чтобы он был недоступен сигналу hang-up**:

```
$ nohup updatedb &              Запускает updatedb с невозможностью
                                его прерывания
# nohup nice -9 gcc hello.c &   Запускает gcc в непрерывном режиме
                                и с более высоким приоритетом
```

Использование команды nohup отличается от запуска команды с использованием символа &, поскольку команда nohup продолжает выполнение команды даже после выхода из запущенной сессии в консоли.

Команда nohup использовалась в то время, когда производительность процессоров была низкой и были распространены соединения по телефонной линии через модем (dial-up) (то есть не было возможности сохранять дорогостоящее соединение в течение длительного времени выполнения команды). Сегодня же, используя такие приложения, как screen (см. гл. 14), вы можете не закрывать консольную сессию, даже если перенесете сетевое подключение в текущую консоль.

Составление расписания запуска процессов

Команды, связанные с утилитой cron, могут использоваться для определения времени автоматического запуска команд (включая настоящий момент времени) в консоли. Команда at **запускает выбранную команду в заданное время**:

```
$ at now +1 min       Запускает команду через одну минуту
at> updatedb
at> <Ctrl+D> <EOT>
job 5 at Mon Aug 20 20:37:00 2007
$ at teatime          Запускает команду в 16.00 текущего дня
$ at now +5 days      Запускает команду через пять дней
$ at 06/25/08         Запускает команду 25 июня 2008 года
                      в текущее время
```

Кроме того, запустить команду, не связанную с консолью, можно с помощью команды batch, которая позволяет настроить запуск команды таким образом, чтобы она запускалась, как только процессор будет готов (средняя нагрузка ниже 8):

```
$ batch         Немедленно запускает команду
at> find /mnt/isos | grep jpg$ > /tmp/mypics
at> <Ctrl+D> <EOT>
```

Стоит отметить, что после выполнения команд at или batch будет отображен следующий запрос at>. Введите команду, которую хотите выполнить, и нажмите клавишу Enter. При необходимости введите дополнительные команды. После завершения для составления списка последовательности выполнения необходимых команд нажмите сочетание клавиш Ctrl+D.

Закончив, вы можете **проверить очередность установленных задач** at, воспользовавшись командой atq:

```
$ atq
11        Wed Sep 5  21:10:00 2007 a francois
10        Fri Aug 24 21:10:00 2007 a francois
8         Thu Aug 23 20:53:00 2007 a francois
```

Обычные пользователи могут просматривать расписание выполнения только своих задач at. Суперпользователь может просматривать все задачи at, находящиеся в очереди. Если вы захотите **удалить задачу из списка очередности**, воспользуйтесь командой atrm:

```
$ atrm 11        Удаляет задачу at под номером 11
```

Команды at и batch используются для установления очередности однократного выполнения команд. Если необходимо **многократно выполнять указанные команды**, воспользуйтесь приложением cron. Эти команды помещаются в список задач cron, а затем в график выполнения, хранящийся в файлах crontab. Существует только один системный файл crontab (/etc/crontab), однако, помимо него, каждый пользователь может создавать индивидуальный файл crontab, предназначенный для запуска команды в назначенное соответствующим пользователем время. Чтобы **создать персональный файл crontab**, выполните следующую команду:

```
$ crontab -e           Создает персональный файл crontab
```

Команда crontab -e, используя текстовый редактор vi, открывает ваш файл crontab (или создает новый). Существуют разные варианты внесения данных в файл crontab:

```
15 8 * * Mon,Tue,Wed,Thu,Fri mail chris < /var/project/stats.txt
* * 1 1,4,7,10 * find / | grep .doc$ > /var/sales/documents.txt
```

В первом примере пользователю **chris** отправляется сообщение, содержащее файл /var/project/stats.txt. Эта команда выполняется ежедневно с понедельника по пятницу включительно в 8:15. Во втором примере в первые дни января, апреля, июля и октября запускается команда find и производит поиск всех имеющихся в системе файлов с расширением DOC, а затем группирует список файлов в файле /var/sales/documents.txt.

В конце каждой строки crontab указывается выполняемая команда. Первые пять полей определяют время и дату запуска команды (поля слева направо: минуты (от 0 до 59), часы (от 0 до 23), дни месяца (от 0 до 31), месяцы (от 0 до 12 или Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov и Dec), а также дни недели (от 0 до 7 или Sun, Mon, Tue, Wed, Thu, Fri и Sat)). Знак * означает, что значение поля не определено и в него может быть добавлена соответствующая информация.

Ниже представлены некоторые другие параметры, которые могут использоваться вместе с командой crontab:

```
# crontab -eu chris            Редактирует файл crontab другого
                               пользователя (только для суперпользователя)
$ crontab -l                   Отображает содержимое файла crontab
15 8 * * Mon,Tue,Wed,Thu,Fri mail chris < /var/project/stats.txt
* * 1 1,4,7,10 * find / | grep .doc$ > /var/sales/documents.txt
$ crontab -r                   Удаляет файл crontab
```

Раньше настройка системных задач cron осуществлялась путем добавления их в системный файл crontab. Хотя данная возможность все еще существует, теперь Ubuntu позволяет настроить ежечасное, ежедневное, еженедельное и ежемесячное выполнение задач cron путем **связывания команды, которую необходимо запустить, с каталогом cron**. Для этого просто введите команду, которую хотите запустить, а затем скопируйте ее в каталог /etc/cron.hourly, /etc/cron.daily, /etc/cron.weekly или /etc/cron.monthly. Команда будет запускаться с периодичностью, соответствующей имени каталога, в котором хранится (ежечасно, ежедневно, еженедельно или ежемесячно).

Утилита anacron является альтернативой утилите cron. Используя anacron, вы можете, как и при использовании cron, настраивать периодичность выполнения команд, однако anacron чаще всего используется на компьютерах, которые не постоянно включены. Если команда не была запущена, поскольку компьютер был выключен во время предполагающегося выполнения команды, то при следующем включении компьютера утилита anacron выполнит заданную команду.

Резюме

Просмотр и работа с процессами, выполняющимися на операционных системах Linux, являются очень важными действиями, обеспечивающими правильную работу системы. Используя такие команды, как ps и top, вы можете просматривать список процессов, запущенных на вашей системе. Кроме того, с помощью команды pgrep вы можете осуществлять поиск и отображать список процессов, удовлетворяющих заданным критериям.

Используя такие команды, как nice и renice, вы можете устанавливать приоритет выполнения процессов. Воспользовавшись во время выполнения процесса командой kill или killall, вы можете изменить выполняемые им действия или уничтожить его.

После запуска команды из текущей консоли вы можете перемещать ее выполнение в фоновый режим (bg) или приоритетный (fg). Вы также можете останавливать и заново запускать процессы, используя контрольные коды.

Для составления графика выполнения команд предназначены команды at или batch. Для составления периодического повторяющегося графика выполнения команд в установленное время предназначены утилиты cron или anacron.

10 Администрирование системы

Без тщательно продуманного управления ожидаемая производительность системы Linux может иногда превышать ее возможности, определяемые доступными ресурсами. При наличии возможности с течением времени контроль системы (использование физической памяти, центрального процессора и устройств) может обеспечить вам уверенность в том, что компьютер способен выполнять необходимые вам задачи. Аналогично контроль над другими аспектами системы, такими как драйверы используемых в ней устройств, может помочь избежать возникновения системных ошибок и проблем с выполнением задач.

Данная глава разбита на несколько разделов, связанных с управлением Ubuntu или другой операционной системой Linux. Первый раздел призван помочь вам научиться отображать объем имеющихся в распоряжении ресурсов (мощность процессора, оперативной памяти и устройств). В следующем разделе описываются способы проверки системного времени. Далее следует описание процесса загрузки и последующего уровня выполнения. В последнем разделе описывается порядок работы с ядром и драйверами устройств, а также получения информации об устройствах, установленных на вашем компьютере.

Приложения для контроля

Ubuntu, Debian и другие операционные системы Linux обладают прекрасным свойством: они контролируют ваши действия. Если вы захотите, то найдете большое количество информации о центральном процессоре, жестких дисках, виртуальной памяти и других ресурсах компьютера, используемых в настоящий момент.

Вы можете в режиме реального времени просматривать информацию, собираемую ядром операционной системы Linux, просто просматривая содержимое файлов файловой системы /proc (см. Приложение 3). Альтернативным способом является использование команд, специально предназначенных для сбора информации об использовании системой виртуальной памяти компьютера, процессора, запоминающих устройств и сетевых интерфейсов.

Существуют команды, способные отображать различные характеристики системных ресурсов. Поскольку данная книга не является простым обзором страниц MAN-справочника, мы разделили следующие разделы по темам (контроль использования оперативной памяти, центрального процессора и запоминающих

устройств), вместо того чтобы осуществлять их разделение по используемым командам (top, vmstat и iostat).

ПРИМЕЧАНИЕ

Некоторые приложения, описываемые в данном разделе, устанавливаются в Ubuntu по умолчанию (вместе с такими пакетами, как procps). Однако, чтобы использовать iostat или sar, вам необходимо установить программный пакет sysstat, для чего выполните следующую команду:

```
$ sudo apt-get install sysstat
```

Оперативная память

Немногие вещи могут так сказаться на производительности системы, как недостаток оперативной памяти. Команды наподобие free и top позволяют просматривать основную информацию об оперативной памяти и файле подкачки. Команда vmstat предоставляет подробную информацию об использовании оперативной памяти и может работать постоянно. Команда slabtop может фиксировать объем использования ядром (слэб-кэш) оперативной памяти.

Команда free предоставляет наиболее быстрый способ просмотра **динамики выделения оперативной памяти** в системе. Она отображает общий объем оперативной памяти (Mem:), объем области подкачки (Swap:), а также объем памяти, используемой в настоящее время. Вот несколько примеров использования команды free:

```
$ free          Отображает объем использования оперативной
                памяти в килобайтах (-k по умолчанию)
            total       used       free     shared    buffers     cached
Mem:       742476     725108      17368          0     153388     342544
-/+ buffers/cache:    229176     513300
Swap:     1020116         72    1020044
$ free -m       Отображает объем использования оперативной памяти в мегабайтах
            total       used       free     shared    buffers     cached
Mem:          725        706         18          0        148        333
-/+ buffers/cache:       223        501
Swap:         996          0        996
$ free -b       Отображает объем использования оперативной памяти в блоках
            total       used       free     shared    buffers     cached
Mem:    760295424  742510592   17784832          0  157114368  350765056
-/+ buffers/cache: 234631168  525664256
Swap: 1044598784      73728 1044525056
$ free -mt      Отображает объем использования и полный объем памяти (Swap + RAM)
            total       used       free     shared    buffers     cached
Mem:          725        708         16          0        149        334
-/+ buffers/cache:       223        501
Swap:         996          0        996
Total:       1721        708       1013
$ free -        Отображает объем использования оперативной памяти в гигабайтах
$ free -s 5     Непрерывно отображает объем используемой
                оперативной памяти (с интервалом в пять секунд)
```

Чтобы избежать недостатка оперативной памяти и ускорения работы приложений, в Linux используется столько незадействованной оперативной памяти, сколько позволяет кэш жесткого диска. Поэтому первую строку отображаемой командой free информации, часто фиксирующую немного свободной оперативной памяти, можно опустить. Мы рекомендуем вам обратить внимание на вторую строку, отображающую общий объем оперативной памяти, доступной для приложений в настоящий момент. В данном примере этот объем равен 501 Мбайт:

```
-/+ buffers/cache:          223         501
```

Одним из способов определить необходимый вашей системе объем оперативной памяти является запуск всех приложений, которые вы планируете запускать одновременно, на другом компьютере с установленной системой Ubuntu. Затем выполните команду free с общим параметром (free -t), чтобы отобразить общий объем используемой памяти, и убедитесь, что на вашей системе общий объем памяти превышает полученное число (желательно, чтобы бо́льшая часть этой памяти приходилась на физическую оперативную память).

Команда top предоставляет средства для просмотра запущенных в настоящий момент процессов и сортирует их по уровню потребления времени центрального процессора или оперативной памяти (для получения более подробной информации обратитесь к гл. 9, в которой описывается использование команды top для отображения запущенных процессов). Вы также можете использовать команду top для **просмотра объема потребляемой оперативной памяти в графическом режиме**:

```
$ top
top - 14:14:59 up 3 days, 18:26, 1 user, load average: 0.11, 0.04, 0.01
Tasks: 114 total, 3 running, 111 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.0%us, 0.0%sy, 0.0%ni,100.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 742476k total, 727232k used, 15244k free, 153708k buffers
Swap: 1020116k total, 72k used, 1020044k free, 343924k cached
  PID USER     PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 2347 root     34  19 89552  77m 5636 S  0.0 10.7   6:05.75 yum-updatesd
 2797 chris    18   0 80612  27m  18m S  0.0  3.8   0:01.29 nautilus
 2814 chris    15   0 44420  22m  20m S  0.0  3.1   0:00.17 nm-applet
```

Для выхода из программы top нажмите клавишу **q**. Как и в выходе команды free, команда top отображает общий объем используемой оперативной памяти (Mem:) и общий объем области подкачки (Swap:). Однако, поскольку top является приложением, ориентированным на экран, и предоставляет возможность непрерывного контроля, вы можете постоянно получать информацию об объеме используемой оперативной памяти, причем каждые три секунды (значение по умолчанию) на экран будет выводиться обновленная информация. Запустив команду top, нажмите сочетание клавиш **Shift+M** — выполняющиеся процессы будут отсортированы по уровню потребления оперативной памяти. Наиболее важным столбцом для определения объема оперативной памяти, потребляемой процессом, является RES, который отображает объем физической памяти, потребляемой процессом и называемой резидентным размером, отображаемым в столбце %MEM.

Для вывода более подробной статистики, касающейся использования виртуальной памяти, используйте команду vmstat. С помощью команды vmstat вы можете

отображать **объем используемой оперативной памяти**, например, со времени последней перезагрузки или за указанный период времени. В следующем примере показано использование команды vmstat, отображающей новую статистику каждые три секунды:

```
$ vmstat 3
procs -----------memory--------- --swap-- ---io---- --system-- -----cpu-----
 r  b   swpd   free  buff  cache  si   so  bi    bo   in   cs us sy id wa st
 1  0  97740  32488  3196 148360   0    0   0     1   26 3876 85 15  0  0  0
 1  1  98388   7428  3204 151472   0  216   0   333   30 3200 82 18  0  0  0
 1  0 113316   8148  2980 146968   0 4980   4  5121   79 3846 77 23  0  0  0
 2  0 132648   7472  2904 148488   0 6455   3  6455   90 3644 83 17  0  0  0
 2  0 147892   8088  2732 144208   0 5085   9  5220   79 3468 84 16  0  0  0
 1  0 157948   7680  2308 134812   0 3272  12  3296   69 3174 77 23  0  0  0
 3  0 158348   7944  1100 123888  21  144  25   275   26 3178 86 14  0  1  0
 2  0 166116   7320   568 120280  11 2401  20  2403   51 3175 84 16  0  0  0
 3  0 181048   7708   648 119452  53 4852 796  4984  123 1783 86 13  0  1  0
```

Для выхода из команды vmstat нажмите сочетание клавиш Ctrl+C. В данном примере vmstat был указан период в 30 секунд, за который было запущено более 100 приложений. Отметим, что, когда объем свободной оперативной памяти находится в пределах от 32 488 Кбайт до 7428 Кбайт (то есть оперативная память перегружена), данные начинают перемещаться в область подкачки (значение 216 в столбце so). Поскольку область подкачки размещена на жестком диске, вы можете видеть, как по мере увеличения области подкачки увеличивается блок, записывающийся на диск (bo). Вы также можете видеть увеличение области подкачки в столбце swpd.

Потребление времени центрального процессора в данном примере также возрастает, но без отображаемого времени ожидания (id 0). Отметим, что, когда в область подкачки необходимо поместить некоторые приложения (последние три строки выхода), процессору для завершения процесса (wa 1) во время ввода/вывода приходится ожидать выполнения двух других процессов.

Вот еще несколько параметров, которые можно использовать с командой vmstat:

```
$ vmstat -S m           Отображает информацию в мегабайтах (1000 Кбайт)
$ vmstat -S M           Выводит информацию в мегабайтах (1024 Кбайт)
$ vmstat -S k           Отображает информацию в килобайтах (1000 байт)
$ vmstat -S K           Выдает информацию в килобайтах (1024 байт)
$ vmstat -n 2 10        Обновляет информацию каждые две секунды в течение
                        20 секунд
$ vmstat -s | less      Отображает счетчик количества событий и статистику
                        использования оперативной памяти
$ vmstat -S M -s | less     Отображает статистику в мегабайтах
      725 M total memory
      717 M used memory
      486 M active memory
      175 M inactive memory
        7 M free memory
        1 M buffer memory
```

```
   120 M swap cache
   996 M total swap
   802 M used swap
   193 M free swap
       ...
```

В предыдущем примере показаны различные варианты отображения статистики использования оперативной памяти (-s) в мегабайтах (-S M), которые, на наш взгляд, наиболее удобны. В остальных примерах продемонстрированы способы отображения информации с помощью команды vmstat в мегабайтах и килобайтах (в упрощенном и техническом исчислении). Параметр -n 2 10 позволяет команде vmstat обновлять результаты через установленный промежуток времени (2) на протяжении заданного количества раз (10).

Используя такие команды, как ps и top, вы можете определять объем используемой вашей системой оперативной памяти. Однако ядро обладает своим собственным кэшем, используемым для контроля ресурсов и называемым *слэбом*. Для **отображения статистики кэш-памяти ядра** воспользуйтесь командой vmstat (из /proc/slabinfo):

```
$ vmstat -m | less    Отображает информацию о кэш-памяти ядра
Cache                Num     Total      Size      Pages
nf_nat:help            2        13       308         13
nf_nat:base            0         0       276         14
bridge_fdb_cache       0         0        64         59
   ...
ext3_inode_cache    1236      2928       488          8
ext3_xattr            29       156        48         78
   ...
```

В информации о кэш-памяти указывается имя каждого объекта, хранящегося в ней, количество активных объектов для данного типа кэша, общее количество объектов, доступных для данного типа кэша, размер кэша (в байтах) и количество страниц каждой области кэша. Используя команду slabtop, вы можете **отобразить информацию о кэш-памяти ядра в графическом режиме** (как и в случае с командой top):

```
$ slabtop
 Active / Total Objects (% used)    : 49127 / 70942 (69.2%)
 Active / Total Slabs (% used)      : 3094 / 3094 (100.0%)
 Active / Total Caches (% used)     : 101 / 145 (69.7%)
 Active / Total Size (% used)       : 8830.29K / 12013.73K (73.5%)
 Minimum / Average / Maximum Object : 0.01K / 0.17K / 128.00K
  OBJS ACTIVE  USE OBJ SIZE  SLABS OBJ/SLAB CACHE SIZE NAME
 11600   4303  37%    0.13K    400       29      1600K dentry_cache
  2928   1246  42%    0.48K    366        8      1464K ext3_inode_cache
  4355   2535  58%    0.28K    335       13      1340K radix_tree_node
   219    219 100%    4.00K    219        1       876K size-4096
  4128   3485  84%    0.16K    172       24       688K filp
```

Выводимая командой slabtop информация обновляется каждые три секунды. По умолчанию содержимое слэб-кэша сортируется по количеству объектов

(первый столбец), хранящихся в каждом разделе кэша. Нажав клавишу C, вы можете отсортировать информацию по размеру кэша (как показано в предыдущем примере).

Процессор

Перегрузка центрального процессора является еще одним важным фактором, который может вызывать сбои в работе системы. Команда vmstat, рассмотренная ранее, может выводить общие статистические данные об использовании процессора (действия пользователя, действия системы, время ожидания, время ожидания сигналов входа/выхода и время, забираемое виртуальной вычислительной машиной). Тем не менее команда iostat (пакет sysstat) может отображать более подробные сведения о загрузке центрального процессора.

Ниже приведено два примера использования команды iostat с целью получения **отчета об использовании процессора**:

```
$ iostat -c 3          Отображает обновляемую каждые три секунды статистику
                       использования процессора (запускаемые приложения)
Linux 2.6.21-1.3194.fc7 (davinci) 08/10/2007
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.50    0.00    0.00    0.00    0.00   99.50
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
          28.71    0.00    5.45   18.32    0.00   47.52
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
          98.99    0.00    1.01    0.00    0.00    0.00
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
          99.50    0.00    0.50    0.00    0.00    0.00
$ iostat -c 3          Отображает обновляемую каждые три секунды статистику
                       использования процессора (копирование файлов)
Linux 2.6.21-1.3194.fc7 (davinci)       08/10/2007
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.50    0.00    0.00    0.00    0.00    0.00
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.50    0.00   24.88   74.63    0.00    0.00
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.50    0.00   10.00   89.50    0.00    0.00
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.50    0.00   17.41   82.09    0.00    0.00
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.00    0.00   14.65   85.35    0.00    0.00
```

В первом примере сначала просто загружается система, а затем начинают выполняться некоторые приложения. Как вы можете заметить, большая часть ресурсов процессора, используемых для работы приложений, задействована в рабочем пространстве пользователя. Во втором примере рассмотрен случай, при котором осуществляется копирование крупных файлов с одного жесткого диска на другой, в результате чего бóльшая часть времени расходовалась на системном уровне, также называемом *пространством ядра*. Отметим, что копирование файлов также влияет на рост времени ожидания завершения запросов ввода/вывода (%iowait).

Ниже приведены примеры использования команды iostat для отображения **отчетов об использовании процессора с применением подписей времени**:

```
$ iostat -c -t          Отображает подпись времени в отчете
                        об использовании процессора
Linux 2.6.21-1.3194.fc7 (davinci)      08/10/2007
Time: 9:28:03 AM
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.50    0.00    0.00    0.00    0.00   99.50
$ iostat -c -t 2        Выводит статистику и обновляет ее
                        каждые две секунды в течение 20 секунд
```

Команда dstat (программный пакет dstat) является альтернативой команде iostat, поскольку также позволяет **просматривать информацию о загрузке центрального процессора** (а также другие данные, связанные с производительностью системы). Преимуществом команды dstat перед другими подобными приложениями является более точное отображение используемых единиц измерения (килобайты или мегабайты) и использование цветов для выделения разных типов данных. Вот пример использования команды dstat для отображения информации о загрузке процессора:

```
$ dstat -t -c 3         Отображает и непрерывно обновляет информацию
                        о загрузке процессора, используя подписи времени
---time--- ----total-cpu-usage----
__epoch___|usr sys idl wai hiq siq
1189727284|  0   0 100   0   0   0
1189727287|  1   0  99   0   0   0
1189727290|  3   0  97   0   0   0
1189727293|  0   0 100   0   0   0
1189727296|  5   0  95   0   0   0
1189727299|  1   0  99   0   0   0
1189727302|  3   0  97   0   0   0
1189727305|  0   0 100   0   0   0
1189727308|  3   0  96   0   1   0
1189727311|  1   0  99   0   0   0
1189727314|  0   0 100   0   0   0
1189727317|  0   0 100   0   0   0
1189727320|  1   0  99   0   0   0
1189727323|  5   0  95   0   0   0
1189727326|  3   0  97   0   0   0
1189727329|  3   0  97   0   0   0
1189727332|  2   0  98   0   0   0
1189727335|  5   0  95   0   0   0
```

В данном случае в выход включены значения даты и времени, основанные на начале времени начала (-t) составления отчета о загрузке процессора (-c), обновляющегося каждые три секунды (3). Этот отчет обновляется до тех пор, пока вы сами не остановите его (нажав сочетание клавиш Ctrl+C).

Если вы хотите определить конкретные процессы, потребляющие наибольший объем ресурсов процессора, воспользуйтесь командой top: введите top, а затем

нажмите Shift+P, чтобы отсортировать процессы по уровню использования процессора (данный порядок сортировки является значением по умолчанию):

```
$ top    Отображает запущенные процессы и сортирует их
         по уровню использования ресурсов процессора
Tasks: 120 total, 3 running, 116 sleeping, 0 stopped, 1 zombie
Cpu(s): 86.8% us, 6.0% sy, 0.0% ni, 3.3% id, 4.0% wa, 0.0% hi, 0.0% si
Mem: 482992k total, 476884k used, 6108k free, 1220k buffers
Swap: 5863716k total, 1166252k used, 4697464k free, 52984k cached
  PID USER     PR  NI VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 9648 chris    16   0 309m 123m  16m R 72.6 26.1 287:55.22 firefox-bin
  552 root     15   0 762m  65m 5732 S 15.6 14.0   4388:27 X
```

Полный список процессов содержит значительно большее количество процессов, отсортированных по степени загрузки процессора (столбец %CPU). В данном примере наибольший объем ресурсов процессора потребляют браузер Firefox (72.6%) и сервер X display (15.6%). Если вы решите завершить процесс Firefox, но по каким-либо причинам не можете корректно закрыть окно Firefox, то введите параметр k, ID процесса Firefox (9648) и номер сигнала (9).

Информацию о самом процессоре вы можете получить прямо из файла /proc/cpuinfo:

```
$ cat /proc/cpuinfo   Отображает информацию о процессоре,
                      содержащуюся в каталоге /proc
processor       : 0
vendor_id       : AuthenticAMD
cpu family      : 6
model           : 4
model name      : AMD Athlon(tm) processor
stepping        : 4
cpu MHz         : 1340.080
cache size      : 256 KB
   ...
flags           : fpu vme de pse tsc msr pae mce cx8 apic mtrr pge mca cmov
pat pse36 mmx fxsr syscall mmxext 3dnowext 3dnow up
bogomips        : 2680.91
clflush size    : 32
```

При описании работы с процессором стоит упомянуть такую небезынтересную деталь, как метки, представляющие собой функции, поддерживаемые процессором. Для работы с некоторыми функциями в Ubuntu необходимо, чтобы были активны определенные расширения процессора, связанные с этими метками. Например, для использования паравиртуальных параметров виртуальной среды Xen должна быть установлена метка pae, а для полного запуска виртуальных параметров — поддержка расширения либо метки vmx (для процессоров Intel), либо svm (для процессоров AMD).

Подобная информация о вашем процессоре собирается системой в самом начале ее загрузки и отображается в начале выхода команды dmesg.

Запоминающие устройства

Основную информацию об объеме памяти, доступной для хранения файловых систем Linux, можно получить, используя такие команды, как du и df (см. гл. 7). Для отображения же более подробной информации о работе запоминающих устройств предназначены команды vmstat и iostat.

Некоторые выходные данные команды iostat, приведенные ранее, могут совпадать с параметрами, используемыми **для определения «узких мест» (аппаратных и программных элементов, ограничивающих производительность системы) при чтении с диска или записи на диск**:

```
$ iostat 3              Проверяет скорости чтения с диска и записи на диск
Linux 2.6.21-1.3194.fc7 (davinci)      08/11/2007
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
          13.15    0.60    0.59    0.16    0.00   85.49
Device:            tps   Blk_read/s   Blk_wrtn/s   Blk_read  Blk_wrtn
sda               1.09        32.08        58.94   16086324  29554312
sdb               0.29         5.27        11.23    2644482   5631348
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           1.00    0.00   42.14   45.15    0.00   11.71
Device:            tps   Blk_read/s   Blk_wrtn/s   Blk_read  Blk_wrtn
sda             411.37     66515.05         2.68     198880         8
sdb              68.23         2.68     14696.99          8     43944
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.67    0.00   41.00   58.33    0.00    0.00
Device:            tps   Blk_read/s   Blk_wrtn/s   Blk_read  Blk_wrtn
sda             239.67     52530.67       106.67     157592       320
sdb             236.00         0.00     55077.33          0    165232
```

В первой части команда iostat отображает уровень средней загрузки центрального процессора со времени последней перезагрузки системы. Далее отображается уровень загрузки процессора при копировании больших объемов данных с первого диска (sda) на второй диск (sdb). Высокие значения iowait свидетельствуют о том, что скорость диска является «узкими местом» системы. Другими словами, увеличение скорости записи на диск приведет к большему приросту производительность системы, нежели увеличение производительности самого процессора.

Команда vmstat также может отображать статистику о ваших дисках. Ниже приведен пример использования команды vmstat **для отображения информации о скорости чтения с диска и записи на него**:

```
$ vmstat -d             Отображает данные о чтении с диска, записи
                        на него и статистику ввода/вывода
disk- ------------reads------------ ------------writes--------      ----IO---
      total merged  sectors     ms   total  merged sectors ms        cur  sec
  ...
sda  32773  74844 19022380 2524211 245477 3473801 29758560 37140075    0 1372
sdb  79963 253716  2646922 2158000  76044  977122  8428140 12489809    0  506
```

В данном примере в системе Linux присутствуют два диска (sda и sdb). Здесь вы можете увидеть общее количество секторов данных жестких дисков, откуда были успешно прочитаны данные и куда была успешно выполнена запись. Кроме того, вы можете увидеть, сколько времени было потрачено на вход/выход (IO) для этих дисков. С помощью следующей команды вы можете просмотреть, выполняются ли в системе какие-либо операции ввода/вывода, а также отобразить информацию о вводе/выводе для выбранных разделов:

```
$ vmstat -p sda1       Отображает статистику чтения/записи
                       для выбранного раздела диска
sda1          reads      read sectors   writes     requested writes
             174060       12993689        2778        22224
```

К сожалению, предыдущая команда не может работать с разделами softraid md, lvm и некоторыми отдельными драйверами устройств RAID.

Если вы хотите определить, какие **файлы и каталоги открыты в данный момент на ваших запоминающих устройствах**, воспользуйтесь командой lsof. Эта команда может быть особенно полезна, если вы хотите размонтировать занятую файловую систему. С помощью этой команды вы можете определить имя открытого и мешающего размонтированию файла, уничтожить процесс, не позволяющий закрыть файл, и затем размонтировать файловую систему. Вот пример использования команды lsof:

```
$ lsof | less  Отображает процессы, не позволяющие закрыть файлы и каталоги
COMMAND     PID  USER    FD    TYPE DEVICE    SIZE      NODE NAME
init          1  root    cwd    DIR    8,5    4096         2 /
init          1  root    rtd    DIR    8,5    4096         2 /
init          1  root    txt    REG    8,5   38620   2049530 /sbin/init
  ...
bash      23857 chris    cwd    DIR    8,1    4096   2719746 /mnt/sda1/dx
```

Первые показанные в примере файлы не позволяют завершить процесс init (первый процесс, запущенный в системе). Файлы, закрыть которые не позволяют системные процессы (например, udevd) и демоны (например, sshd и syslogd), привязаны к процессу init. В конце списка отображаются файлы, закрыть которые не позволяют отдельные пользователи (заинтересованные, возможно, в том, чтобы вы не могли размонтировать раздел диска).

ПРИМЕЧАНИЕ

Если вы не выполните команду $ sudo lsof | less, то можете столкнуться с ограничениями прав доступа.

После отображения информации с помощью команды lsof вам, возможно, захочется определить имя каталога или файла (NAME), команду, открывшую их (COMMAND), и ID процесса этой запущенной команды (PID). Поскольку случай, когда вы не можете размонтировать файловую систему, является довольно частым, в предыдущем примере файловая система /mnt/sda1 была открыта консолью bash (/mnt/sda1/dx является текущим рабочим каталогом консоли bash). На самом деле вместо перенаправления информации, возвращаемой командой lsof, в less или grep суще-

ствует несколько других способов, позволяющих отобразить дополнительную информацию с помощью lsof:

```
$ lsof -c bash              Отображает файлы, используемые консолью bash
$ lsof -d cwd               Выводит каталоги, открытые как текущие рабочие
                            каталоги
$ lsof -u chris             Отображает файлы и каталоги, открытые
                            пользователем chris
$ lsof /mnt/sda1            Выдает все открытые элементы файловой системы
                            /mnt/sda1
$ lsof +d /mnt/sda1/dx      Отображает все открытые элементы, хранящиеся
                            в каталоге /mnt/sda1/dx и его подкаталогах
```

Как было отмечено ранее, вам может понадобиться использовать команду sudo для получения прав суперпользователя, необходимых для просмотра информации команды lsof.

Управлением временем

Соблюдение необходимых временных режимов в операционной системе Linux является важным фактором для обеспечения правильной работы системы. На компьютерах с операционной системой Linux используется два вида времени: системное время (используется в Linux для соблюдения временных режимов) и аппаратное время (устанавливает системное время при запуске Linux).

Системное время используется для установки подписей времени при создании файлов, в рабочем цикле процессов, а также во всех остальных параметрах, в которых задействуются понятия даты и времени. Системное время доступно для просмотра и может устанавливаться вручную (с помощью команды date) или автоматически (с помощью сервиса ntpd).

Аппаратное время является CMOS-составляющей материнской платы и поддерживается с помощью батареи, расположенной на материнской плате, когда система отключена. Устанавливать аппаратное время можно с помощью команды hwclock.

В системах Linux существует много приложений, позволяющих работать со временем. Например, существуют утилиты для проверки времени различными способами: команда clockdiff (используется для определения разницы в настройке системных часов двух компьютеров) и uptime (отображает продолжительность работы системы).

Изменение даты/времени с помощью графических утилит

Среди графических средств, использующихся в Ubuntu и других операционных системах Linux, для изменения даты, времени и временно́й зоны можно назвать окно **Date and Time Settings** (Настройка даты и времени) (команда sudo time-admin). Это окно также может быть использовано для включения сетевого протокола времени (NTP) с целью автоматической синхронизации по сети даты и времени

операционной системы с выбранным сервером времени. Для этого может потребоваться поддержка NTP, доступ к которой осуществляется нажатием кнопки меню окна **Date and Time Settings** (Настройка даты и времени).

Окно **Date and Time Settings** (Настройка даты и времени) сохраняет все производимые изменения. Во время загрузки Ubuntu система считывает эти настройки для правильной установки временно́й зоны и времени UTC (если используется).

Временная зона для операционной системы Linux устанавливается на основе данных файла /etc/localtime. Вы можете в любой момент поменять временну́ю зону, скопировав файл с настройками новой временно́й зоны /usr/share/zoneinfo. Например, чтобы изменить текущую временну́ю зону на зону США/Чикаго (America/Chicago), выполните следующую команду:

```
$ sudo cp /usr/share/zoneinfo/America/Chicago /etc/localtime
```

Кроме того, можно воспользоваться командой symlink:

```
$ sudo ln -s /usr/share/zoneinfo/America/Chicago /etc/localtime
```

Для временного изменения временно́й зоны нужную зону также можно выбрать в окне **Date and Time Settings** (Настройка даты и времени).

Отображение и установка системного времени

Команда date, если отсутствуют автоматические настройки NTP, является первичным консольным интерфейсом для отображения и проверки на́строек даты и времени.

Ниже приведены примеры использования команды date **для отображения различными способами даты и времени**:

```
$ date                             Отображает текущую дату, время
                                   и временную зону
Sun Aug 12 01:26:50 CDT 2007
$ date '+%A %B %d %G'              Отображает день недели, месяц, число, год
Sunday August 12 2007
$ date '+The date today is %F.     Добавляет текст к дате
The date today is 2007-08-12
$ date --date='4 weeks'            Отображает дату, которая наступит через
                                   четыре недели (начиная с сегодняшнего дня)
Sun Sep 9 10:51:18 CDT 2007
$ date --date='8 months 3 days'    Отображает дату, которая наступит через
                                   восемь недель и три дня (начиная
                                   с сегодняшнего дня)
Tue Apr 15 10:59:44 CDT 2008
$ date --date='4 Jul' +%A          Отображает день недели, на который
                                   приходится 4 июля
Wednesday
```

Хотя в данном разделе первостепенным является понятие времени, поскольку мы остановились на работе с датой, приведем несколько примеров использования команды cal, предоставляющей очень быстрый способ **отобразить даты по месяцам**.

```
$ cal     Отображает текущий месяц в виде календаря (сегодняшний день выделен)
   August 2007
Su Mo Tu We Th Fr Sa
          1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31
$ cal 2007     Отображает календарь на весь год
                                 2007
           January                  February                 March
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
    1  2  3  4  5  6                1  2  3                1  2  3
 7  8  9 10 11 12 13    4  5  6  7  8  9 10    4  5  6  7  8  9 10
14 15 16 17 18 19 20   11 12 13 14 15 16 17   11 12 13 14 15 16 17
21 22 23 24 25 26 27   18 19 20 21 22 23 24   18 19 20 21 22 23 24
28 29 30 31            25 26 27 28            25 26 27 28 29 30 31
...
$ cal -j Отображает юлианский календарь (начало с 1 января)
August 2007
Sun Mon Tue Wed Thu Fri Sat
            213 214 215 216
217 218 219 220 221 222 223
224 225 226 227 228 229 230
231 232 233 234 235 236 237
238 239 240 241 242 243
```

Команда date также может использоваться **для изменения системной даты и времени**:

```
$ sudo date 081215212008          Изменяет дату и время на 12 Авг. 14:21 2008
Tue Aug 12 11:42:00 CDT 2008
$ sudo date --set='+7 minutes'    Переводит часы на 7 минут вперед
Sun Aug 12 11:49:33 CDT 2008
$ sudo date --set='-1 month'      Изменяет текущий месяц на предыдущий
Sun Jul 12 11:50:20 CDT 2008
```

При следующей загрузке Ubuntu произойдет сброс системного времени, и оно будет установлено в соответствии с новыми настройками аппаратного времени (или в соответствии с данными сервера NTP, если он доступен). В следующий раз после выключения компьютера будет произведен сброс аппаратного времени до значения системного времени, чтобы сохранить его значение, пока компьютер будет выключен. Для изменения аппаратного времени можно использовать команду hwclock.

Отображение и настройка аппаратного времени

Любой пользователь с помощью команды hwclock может просматривать настройки аппаратного времени, однако для изменения данных настроек необходимо обладать

привилегиями суперпользователя. Чтобы **определить текущее аппаратное время компьютера**, выполните следующее:

```
$ hwclock -r            Отображает текущие настройки аппаратного времени
Sun 12 Aug 2007 03:45:40 PM CDT -0.447403 seconds
```

Даже если аппаратное время настроено в режиме времени UTC, команда hwclock по умолчанию отображает местное время. Если системное время отличается от аппаратного (например, при попытке выполнения одной из команд date, описанных ранее), то можно **сбросить системное время до значения аппаратного**:

```
$ sudo hwclock -hctosys     Сбрасывает системное время до значения аппаратного
```

Аналогично, если аппаратное время настроено некорректно (например, из материнской платы была извлечена батарея CMOS), можно **синхронизировать аппаратное время с системным**:

```
# hwclock -systohc      Сбрасывает аппаратное время до значения системного
```

С течением времени значение аппаратного времени может сбиваться. Поскольку отклонение происходит каждый день на одинаковое значение, hwclock может выравнивать его (в файле /etc/adjtime). **Выровнять значение аппаратного времени** с помощью файла adjtime можно следующим образом:

```
$ sudo hwclock -adjust       Задает смещение значения аппаратного времени
```

Для **установки конкретного значения аппаратного времени** предназначен параметр --set:

```
$ sudo hwclock --set --date="3/18/08 18:22:00" Устанавливает новые значения
                                               даты и времени
```

В этом примере аппаратное время изменяется на 18 часов 22 минуты 18 марта 2008 года. Это обновление не сразу сказывается на значении системного времени.

Использование сетевого протокола времени для установки даты и времени

При установке Ubuntu пользователю предоставляется возможность указать текущую дату и время. На данном этапе можно выбрать параметр, позволяющий использовать заранее настроенные серверы сетевого протокола времени (NTP) для автоматической синхронизации времени и даты при перезагрузке системы. При выборе соответствующего параметра программа-установщик установит пакет ntpd, который в дальнейшем будет использоваться для синхронизации.

Если при установке операционной системы Linux сервер NTP настроен не был, можно настроить его позже, **задействовав сервис ntpd**. Чтобы установить данный сервис, нажмите кнопку меню окна **Date and Time Settings** (Настройка даты и времени) либо выполните соответствующую команду.

Ниже приведена команда, позволяющая активировать данный сервис из командной строки:

```
$ sudo apt-get install ntp  Устанавливает пакет ntp (при необходимости)
                            и запускает сервис
```

Сервис ntpd использует информацию, содержащуюся в файле /etc/ntpd.conf. Чтобы, к примеру, формировать запросы о предоставлении информации о времени, укажите имя или IP-адрес сервера времени.

Независимо от того, каким образом устанавливается утилита ntpd — вручную или автоматически, компьютер подключается к серверу времени, устанавливая соединение с UDP-портом 123. Однако, если в этом нет особой необходимости (а также отсутствует собственный GPS или квантовые часы), запуск ntpd может чрезмерно загрузить систему и поставить под угрозу ее безопасность. Именно по этой причине некоторые системные администраторы для настройки времени системы через NTP предпочитают использовать команду ntpdate (зачастую в ежедневном планировщике cronjob):

```
$ sudo ntpdate pool.ntp.org
15 Aug 00:37:12 ntpdate[9706]:
adjust time server 66.92.68.11 offset 0.009204 sec
```

Если вы попытаетесь запустить команду ntpdate одновременно с ntpd, будет выдано сообщение об ошибке:

```
$ sudo ntpdate pool.ntp.org
15 Aug 00:37:00 ntpdate[9695]: the NTP socket is in use, exiting
```

Отметим, что команда ntpdate уже была выделена и в будущем ее использование будет прекращено. Она будет заменена следующими параметрами ntpd:

```
$ sudo ntpd -qg
```

Параметр -q указывает утилите ntpd прекратить работу после синхронизации времени (при работе в качестве демона она продолжает работать). Параметр -g предотвращает выполнение ntpd, если системные часы отключены на протяжении более 1000 секунд.

Управление процессом загрузки

При первом включении компьютера базовая система ввода/вывода (BIOS) просматривает настройки очереди загрузки для определения операционной системы, которая должна быть загружена. Обычно, если загрузочный носитель не вставлен в привод (CD, DVD, дискета и т. д.), BIOS просматривает главную загрузочную запись (MBR) первого загрузочного жесткого диска. В большинстве систем Linux контроль процесса загрузки с данного этапа осуществляется *загрузчиком.*

В Ubuntu и большинстве современных операционных систем Linux загрузчиком по умолчанию является основной общий загрузчик (GRUB). GRUB заменяет LILO, который был в 1990-е годы основным загрузчиком Linux. GRUB можно настроить таким образом, чтобы он загружал не только одну операционную систему Linux, но и другие операционные системы, установленные на жестких дисках (Windows, BSD и др.). Для более тонкой настройки процесса загрузки GRUB

поддерживает использование параметров загрузки каждой загружаемой операционной системы: например, включение или выключение поддержки отдельных устройств.

После выбора в загрузчике операционной системы Linux загружается ядро. При этом возникает следующая проблема: ядру необходимо смонтировать корневую файловую систему на жесткий диск, а для этого необходимы соответствующие драйверы накопителей (модули ядра блочных устройств), которые размещаются на жестком диске. Для выхода из этого замкнутого круга загрузчиком монтируется небольшой псевдодиск из оперативной памяти (initrd), содержащий модули блочных устройств. Это позволяет ядру Linux осуществлять чтение корневой файловой системы. После этого загружается процесс init, который, в свою очередь, запускает системные службы, основываясь на уровне выполнения, задаваемом системой.

В следующем разделе описываются команды, предназначенные для настройки загрузчика, коды запуска и уровни выполнения, свойственные данному типу операционной системы Linux.

Загрузчик GRUB

Предположим, что GRUB был установлен при первом запуске Ubuntu, а настройки для загрузчика были помещены в файл /boot/grub/menu.lst. Любые изменения, вносимые в этот файл, автоматически применяются при перезагрузке Ubuntu. Ниже представлен пример содержимого файла /boot/grub/menu.lst:

```
default=0
timeout=5
hiddenmenu
title Ubuntu, kernel 2.6.20-16-generic
root (hd0,0)
kernel          /vmlinuz-2.6.20-16-generic
root=UUID=db2dac48-a62e-4dbe-9529-e88a57b15bac ro quiet splash
initrd          /initrd.img-2.6.20-16-generic
```

В данном примере отображена информация только об одной загружаемой операционной системе (Ubuntu), хотя в файле по умолчанию обычно предлагаются и такие варианты, как резервная версия Ubuntu. Строка default=0 сообщает о том, что первая строка загружается по умолчанию. Строка timeout=5 означает, что перед началом загрузки GRUB выводит на пять секунд экран-заставку. Строка hiddenmenu определяет, что во время отображения заставки не будет выводиться список загружаемых компонентов (для открытия меню достаточно нажать любую кнопку).

Запись загрузки (title Ubuntu) относится к первому разделу первого жесткого диска (hd0,0), на котором находятся загружаемые ядро и начальный псевдодиск (initrd). Для изменения способа загрузки ядра достаточно добавить соответствующие параметры в конец строки ядра. Для загрузки же различных ядер или операционных систем можно добавить полностью новые описания.

Некоторые возможные параметры загрузки представлены в табл. 2.1. Среди параметров, которые можно добавить в конец строки ядра, можно отметить 3 (за-

грузка с уровнем выполнения 3 и в текстовом режиме) и ide=nodma (отключение DMA, если в системе присутствуют ошибки жесткого диска).

При нормальных обстоятельствах для добавления файла menu.lst в загрузчик не требуется выполнять каких-либо команд. При перезагрузке файл menu.lst считывается напрямую с жесткого диска, однако, если MBR по каким-либо причинам перестанет работать и система не будет загружаться, может понадобиться перезагрузить загрузчик GRUB.

Чтобы переустановить GRUB на MBR жесткого диска, необходимо загрузить в безопасном режиме Live CD или установочный компакт-диск Ubuntu и следовать инструкциям по смене суперпользователя (chroot) для раздела жесткого диска, в который установлена система Ubuntu. Теперь предположим, что загрузка произошла с первого жесткого диска SATA. Тогда, чтобы перезагрузить загрузчик для MBR, необходимо выполнить следующую команду:

```
$ sudo grub-install /dev/sda
```

Теперь загрузчик должен установиться на MBR жесткого диска, и если с файлом menu.lst все в порядке, операционная система загрузится с жесткого диска.

Кроме этого, для создания файла menu.lst достаточно выполнить команду update-grub, а затем следовать отображаемым инструкциям:

```
$ sudo update-grub
```

справление неполадок основного :севдодиска (initrd)

Файл initrd хранится в папке /boot под таким именем: initrd.img-2.6.20-16-generic. Если файл initrd будет поврежден или потребуется добавить в него новые драйверы блочных устройств, воспользуйтесь командой mkinitrd. Сначала сделайте копию старого файла, а затем выполните следующие команды:

```
$ sudo apt-get install initrd-tools
$ sudo mkinitrd -o /boot/ initrd.img-2.6.20-16-generic
```

Теперь замените ядро старой версии (2.6.20-1.2320.fc5) новым. Чтобы использовать предыдущую версию ядра, выполните следующее:

```
$ sudo mkinitrd -o /boot/initrd.img-`uname -r` `uname -r`
```

К сожалению, часто пользователи слишком поздно осознают, что им нужно заменить файл initrd, — после того как замечают неполадки в ядре во время монтирования корневой файловой системы. Если это произойдет, перезагрузите систему в безопасном режиме, как это было описано ранее, а затем выполните команды chrooting и mkinitrd для соответствующего раздела жесткого диска.

онтроль уровней загрузки и выполнения

После загрузки ядра контроль над системой осуществляется процессом init, который является первым запускаемым процессом в системе (PID 1) и управляет загрузкой остальных процессов на основе данных, содержащихся в файле /etc/

inittab, уровнем выполнения по умолчанию и командными строками init, которые выполняются на данном уровне.

Уровень выполнения по умолчанию обычно устанавливается в значение 5 для настольных систем и 3 — для серверов (основываясь на значении команды telinit в файле /etc/event.d/rc-default). Как было отмечено ранее, это значение может быть изменено посредством присвоения ему другой величины (S, 1, 2, 3, 4 или 5) в конце строки ядра на экране загрузки.

Большинство администраторов Linux сохраняют основные параметры загрузки по умолчанию и концентрируются на сервисах, включаемых или отключаемых при выбранном уровне выполнения. Механизм запуска исполняемых фалов при заданном уровне выполнения в Ubuntu и подобных системах основывается на утилите System V (программные пакеты sysvinit и initscripts), изначально использовавшейся в системах AT&T и UNIX System V.

ПРИМЕЧАНИЕ

Несмотря на то что Ubuntu вместо init использует upstart, уровни выполнения можно изменять с помощью команд init или telinit.

В данном разделе речь пойдет о работе с начальными командными строками системы и уровнями их выполнения. Будучи суперпользователем, для **просмотра текущего уровня выполнения** можно выполнить команду runlevel:

```
$ runlevel                Отображает текущие и предыдущие уровни выполнения
N 3
```

Поскольку в данном примере система загружалась с уровнем выполнения 3, предыдущего уровня выполнения просто не существует (N). Чтобы **изменить текущий уровень выполнения**, необходимо использовать команду init:

```
$ sudo init 5   Изменяет текущий уровень выполнения на 5 (Рабочий стол X)
```

В данном примере уровень выполнения изменяется с предыдущего (в данном случае 3) на уровень выполнения 5 (при котором запускается графический интерфейс пользователя X Window). Кроме того, **чтобы заново проверить файл /etc/inittab и остановить или запустить процессы с учетом внесенных в данный файл изменений**, можно воспользоваться командой init вместе с параметром q:

```
$ sudo init q   Запускает или останавливает процессы, измененные
                в файле inittab
```

Отметим, что, если выполнить команду init q, службы System V не будут остановлены. Данная команда чаще всего используется для внесения поверхностных исправлений в файл gettys каталога /etc/event.d.

Для управления сервисами можно использовать команды chkconfig и service. Например, чтобы **немедленно запустить сервис NTP**, выполните следующую команду:

```
$ sudo /etc/init.d/ntp start      Немедленно запускает сервис Samba
Starting NTP server ntpd    [  OK  ]
```

Каждый сервис имеет консольную командную строку в каталоге /etc/init.d. Для любого сервиса можно применять параметры start или stop:

```
sudo /etc/init.d/service_to_control start
sudo /etc/init.d/service_to_control stop
```

Практически все командные строки, хранящиеся в каталоге /etc/init.d, поддерживают параметры start и stop, однако некоторые поддерживают еще и дополнительные параметры. Ниже приведен пример использования сервисов **для запуска и завершения процессов**:

```
$ /etc/init.d/ntp              Отображает статистику использования
                               (без параметров)
Usage: /etc/init.d/ntp {start|stop|restart|try-restart|force-reload|status}
$ sudo /etc/init.d/ntp restart      Перезапускает сервис NTP (выключает
                                    и затем включает)
* Stopping NTP server ntpd          [ OK ]
* Starting NTP server ntpd          [ OK ]
$ sudo /etc/init.d/ntp try-restart  Перезапускает сервис NTP
                                    (если уже запущен)
* Stopping NTP server ntpd          [ OK ]
* Starting NTP server ntpd          [ OK ]
$ sudo /etc/init.d/ntp force-reload Обновляет настройки конфигурационного
                                    файла
* Stopping NTP server ntpd          [ OK ]
* Starting NTP server ntpd          [ OK ]
$ sudo /etc/init.d/ntp status       Проверяет, запущен ли сервис NTP
* NTP server is running.
$ sudo /etc/init.d/ntp stop         Останавливает сервис NTP
* Stopping NTP server ntpd          [ OK ]
```

Любые командные строки init, содержащиеся в файле /etc/init.d, могут быть запущены таким образом, но не все они поддерживают указанные параметры. Однако большинство командных строк init отобразят статистику об их использовании без параметров (как показано в первом примере выше).

Хотя предыдущие команды немедленно запускают командную строку уровня выполнения, для автоматического запуска сервиса во время загрузки или изменения уровня выполнения можно использовать команду update-rc.d. Кроме того, большинство установочных командных строк для сервисов автоматически включают сервис при следующей загрузке. С помощью команды update-rc.d можно включать службы или выключать их, основываясь на уровне их выполнения, например:

```
$ sudo update-rc.d ntp defaults          Включает службу NTP
```

ПРИМЕЧАНИЕ

Справочная документация к команде update-rc.d рекомендует не использовать данную команду для управления уровнями выполнения. Для получения более подробной информации обратитесь к разделу MAN-руководства, посвященному команде update-rc.d.

Хотя для изменения любого уровня выполнения, включая init 0 (выключение) и init 6 (перезагрузка), можно использовать команду init, для остановки работы Linux предусмотрены специальные команды.

Преимуществом таких команд, как halt, reboot, poweroff и shutdown, является наличие в них параметров для **остановки некоторых приложений перед выключением системы**:

ВНИМАНИЕ

Не используйте следующие команды, если не собираетесь завершить работу системы (особенно если эта система удаленная).

```
$ sudo reboot                Перезагружает компьютер
$ sudo halt -n               Останавливает запуск sync для синхронизации
                             жестких дисков перед выключением системы
$ sudo halt -h               Перед остановкой работы системы переводит
                             жесткие диски в режим ожидания
$ sudo shutdown 10           Предупреждает пользователей и через десять минут
                             выключает компьютер
$ sudo shutdown -r 10        Предупреждает пользователей и через десять минут
                             перезагружает компьютер
$ sudo shutdown 10 'Bye!'    Отправляет каждому пользователю сообщение перед
                             выключением компьютера
```

Помимо команд reboot и init 6, для перезагрузки компьютера можно использовать стандартное сочетание клавиш Ctrl+Alt+Delete.

Ядро

Во время загрузки операционной системы Linux с ее ядром можно выполнять много различных операций. Существуют специальные приложения для просмотра состояния используемого ядра и информации о процессе его загрузки. Кроме того, существуют приложения, используемые для обеспечения дополнительной поддержки ядра и в случаях, если что-то идет не так.

Для **определения того, какое ядро запущено в системе в настоящий момент**, необходимо выполнить следующую команду:

```
$ uname -r      Отображает версию ядра
2.6.20-16-generic
$ uname -a      Отображает всю доступную информацию о ядре
Linux server.domain.com #2 SMP Fri Aug 31 00:55:27 UTC 2007 i686 GNU/Linux
```

При запуске ядра сообщения о происходящих событиях помещаются в кольцевой буфер ядра. Чтобы **отобразить содержимое кольцевого буфера ядра**, воспользуйтесь командой dmesg:

```
$ dmesg |less
[    0.000000] Linux version 2.6.20-16-generic (root@terranova) (gcc version
4.1.2 (Ubuntu 4.1.2-0ubuntu4)) #2 SMP Fri Aug 31 00:55:27 UTC 2007 (Ubuntu
2.6.20-16.31-generic)
[    0.000000] BIOS-provided physical RAM map:
```

```
[    0.000000] sanitize start
[    0.000000] sanitize end
   ...
[   15.935761] CPU: L1 I cache: 32K, L1 D cache: 32K
[   15.935763] CPU: L2 cache: 2048K
[   15.935765] CPU: Physical Processor ID: 0
[   15.935767] CPU: Processor Core ID: 0
[   15.935768] CPU: After all inits, caps: bfebfbff 20100000 00000000
00003940 0000e3bd 00000000 00000001
```

Если буфер переполнится, то начало информации может быть потеряно. В этом случае используйте команду less/var/log/dmesg.

Дополнительную информацию о функционировании ядра можно найти в файлах /var/log, в частности в файлах сообщений. Эти файлы можно просмотреть следующим образом:

```
$ sudo cat /var/log/messages* | less      Отображает сообщения /var/log/
Aug 5 21:55:46 davinci syslogd 1.4.2: restart.
Aug 6 22:12:03 davinci kernel: eth0: link up, 100Mbps, lpa 0x45E1
Aug 6 22:13:06 davinci kernel: eth0: link down
Aug 6 22:13:07 davinci kernel: eth0: link up, 100Mbps, lpa 0x45E1
Aug 10 10:53:46 davinci init: Switching to runlevel: 3
```

В идеальном случае должны быть найдены и определены все присоединенные к компьютеру устройства и для них должны быть установлены соответствующие драйверы. Однако в некоторых случаях может быть определен неверный драйвер либо необходимый драйвер может быть недоступен на системе. Для таких случаев в Linux предусмотрена возможность просмотра загружаемых и добавления новых модулей ядра в систему.

Команда lsmod позволяет **просматривать имена загружаемых модулей**, их размер, а также отображать использующие их модули:

```
$ lsmod
Module                 Size Used by
  ...
parport_pc            29797 1
parport               38025 2 lp,parport_pc
snd_ens1371           28769 1
gameport              19017 1 snd_ens1371
snd_rawmidi           26561 1 snd_ens1371
snd_ac97_codec        96357 1 snd_ens1371
ac97_bus               6465 1 snd_ac97_codec
snd_timer             24773 2 snd_seq,snd_pcm
soundcore             11553 2 snd
e100                  37193 0
```

Если вы захотите **получить более подробную информацию о конкретном модуле**, воспользуйтесь командой modinfo:

```
$ modinfo snd_ens1371
filename:    /lib/modules/2.6.21-1.3194.fc7/kernel/sound/pci/snd-ens1371.ko
description: Ensoniq/Creative AudioPCI ES1371+
```

```
license:    GPL
author:     Jaroslav Kysela <perex@suse.cz>, Thomas Sailer
            <sailer@ife.ee.ethz.ch>
srcversion: 411FDA312BD30C6B2A8F6E7
alias:      pci:v00001102d00008938sv*sd*bc*sc*i*
alias:      pci:v00001274d00005880sv*sd*bc*sc*i*
alias:      pci:v00001274d00001371sv*sd*bc*sc*i*
depends:    snd-pcm,snd,snd-rawmidi,gameport,snd-ac97-codec
vermagic:   2.6.21-1.3194.fc7 SMP mod_unload 686 4KSTACKS
parm:       index:Index value for Ensoniq AudioPCI soundcard. (array of int)
parm:       id:ID string for Ensoniq AudioPCI soundcard. (array of charp)
parm:       enable:Enable Ensoniq AudioPCI soundcard. (array of bool)
parm:       joystick_port:Joystick port address. (array of int)
```

Если потребуется **добавить или удалить загружаемый модуль**, чтобы некоторые устройства работали правильно, используйте команду modprobe. Команду modprobe также можно использовать **для отображения списка всех доступных модулей или удаления некоторых из них**:

```
$ modprobe -l | grep c-qcam        Отображает все модули, затем ищет c-qcam
/lib/modules/2.6.21-1.3228.fc7/kernel/drivers/media/video/c-qcam.ko
$ sudo modprobe c-qcam             Загружает модуль Color QuickCam
$ sudo modprobe -r c-qcam          Удаляет модуль Color QuickCam
```

ПРИМЕЧАНИЕ

Возможно, вы слышали о команде insmod. Команда insmod для modprobe представляет то же самое, что и команда rpm для apt-get: modprobe может интеллектуально осуществлять загрузку зависимостей модулей. Именно поэтому мы рекомендуем использовать только modprobe.

Используя команду sysctl, вы можете контролировать параметры ядра при запущенной системе. Кроме того, с ее помощью можно добавлять параметры в файл /etc/sysctl.conf, если необходимо, чтобы они загружались одновременно как группа или при каждой перезагрузке системы:

```
$ sudo sysctl -a | less            Отображает все параметры ядра
kernel.panic = 0
kernel.exec-shield = 1

$ sudo sysctl kernel.hostname      Отображает значения отдельных параметров ядра
$ sudo sysctl -p                   Загружает параметры из файла /etc/sysctl.conf
$ sudo sysctl -w kernel.hostname=joe Устанавливает значение kernel.hostname
```

Как было отмечено ранее, если необходимо изменить параметры ядра, то их можно добавить в файл /etc/sysctl.conf. Настройки параметров в данном файле имеют следующий вид: параметр = значение.

Запись устройств по машинному адресу

Если необходимо получить подробную информацию об устройствах компьютера, попробуйте воспользоваться одной из следующих команд. Команда lspci предна-

значена для **отображения информации об устройствах PCI**, установленных на компьютере:

```
$ lspci          Отображает список устройств PCI
00:00.0 Host bridge: VIA Technologies, Inc. VT8375 [KM266/KL266] Host Bridge
00:01.0 PCI bridge: VIA Technologies, Inc. VT8633 [Apollo Pro266 AGP]
00:10.0 USB Controller: VIA Technologies, Inc. VT82xxxxx UHCI USB 1.1
00:11.0 ISA bridge: VIA Technologies, Inc. VT8235 ISA Bridge
00:12.0 Ethernet controller: VIA Technologies, Inc. VT6102 [Rhine-II]
01:00.0 VGA compatible controller: S3 Inc. VT8375 [ProSavage8 KM266/KL266]

$ lspci -v          Отображает список устройств PCI (более подробно)
$ lspci -vv         Отображает список устройств PCI (еще более подробно)
```

Используя команду dmidecode, можно **отображать информацию об устройствах компьютера**, включая информацию о поддерживаемых функциях BIOS:

```
$ sudo dmidecode | less          Отображает список устройств
$ sudo dmidecode 2.7
SMBIOS 2.3 present.
32 structures occupying 919 bytes.
Table at 0x000F0100.
Handle 0x0000, DMI type 0, 20 bytes.
BIOS Information
 Vendor: Award Software International, Inc.
 Version: F2
 Release Date: 10/06/2003

Processor Information
 Socket Designation: Socket A
 Type: Central Processor
 Family: Athlon
 Manufacturer: AMD
 ID: 44 06 00 00 FF FB 83 01
 Signature: Family 6, Model 4, Stepping 4
 Flags:
        FPU (Floating-point unit on-chip)
        VME (Virtual mode extension)
        DE (Debugging extension)
```

Для **просмотра и изменения информации о жестком диске** предназначена команда hdparm.

ВНИМАНИЕ

Хотя сама по себе данная команда не представляет опасности, она может потенциально повредить жесткий диск, если изменить некоторые из параметров.

Вот несколько примеров отображения информации о жестких дисках:

```
$ sudo hdparm /dev/sda       Отображает параметры жестких дисков (SATA или SCSI)
/dev/sda:
 IO_support = 0 (default 16-bit)
```

```
 readonly = 0 (off)
 readahead = 256 (on)
 geometry = 30401/255/63, sectors = 488395055, start = 0
$ sudo hdparm /dev/hda      Отображает параметры жесткого диска (IDE)
$ sudo hdparm -I /dev/sda   Выводит подробную информацию о жёстком диске
/dev/sda:
ATA device, with non-removable media
        Model Number: FUJITSU MPG3409AT E
        Serial Number: VH06T190RV9W
        Firmware Revision: 82C5
```

Резюме

Ubuntu и другие операционные системы Linux предоставляют простые инструменты просмотра и изменения многих параметров запущенной системы, позволяющие убедиться, что операционная система работает с наилучшей производительностью. Такие команды, как free, top, vmstat, slabtop, iostat и dstat, позволяют просматривать информацию об использовании системой ресурсов центрального процессора, оперативной памяти и запоминающих устройств. Используя команды date, hwclock и cal, а также сервисы типа NTP, можно управлять системными настройками даты и времени.

Для управления установленными функциями и сервисами, запускающимися при загрузке системы, можно изменять параметры, связанные с загрузчиком GRUB и системными уровнями выполнения. Используя такие команды, как service и chkconfig, можно запускать, останавливать, отображать, добавлять и удалять отдельные системные сервисы. Команды reboot, halt и shutdown позволяют безопасно выключать и перезагружать компьютер.

Когда приходит время управлять устройствами компьютера, такие команды, как lsmod, modinfo и modprobe, дают возможность работать с загружаемыми модулями. Используя команды lspci, dmidecode и hdparm, вы можете просматривать информацию об устройствах компьютера.

11 Управление сетевыми подключениями

Подключение компьютера с операционной системой Linux к сети часто осуществляется посредством подключения карты сетевого интерфейса к устройству ISP (например, DSL или кабельному модему) и перезагрузки. Однако, если сетевой интерфейс отказывается работать или требует ручной настройки, можно воспользоваться специальными командами, предназначенными для настройки сети, проверки сетевых соединений и настройки специальной маршрутизации.

В данной главе описываются команды для настройки и работы с картами сетевого интерфейса (NICs), например `ethtool`, `mii-tool` и `ifconfig`. В частности, здесь описываются способы настройки современных сетевых устройств, а также проводной и беспроводной сетей Ethernet. В данной главе описываются такие команды, как `netstat`, `dig`, `ip` и `ping`, предназначенные для получения информации о сети (необходимо иметь подключенные сетевые устройства).

астройка сетей с помощью GUI

При первой установке Ubuntu программа-установщик позволяет настроить любые сетевые карты Ethernet, подключенные к компьютеру, с помощью сервера DHCP, определенного в данной сети. Кроме того, наряду с именем хост-системы и IP-адресами машин-шлюзов и серверами имен, программа помогает настроить стационарный IP-адрес. После установки системы для настройки сетевых интерфейсов можно использовать графические приложения.

Окно Network Configuration (Настройки сети) (меню GNOME ▸ System ▸ Administrat ion ▸ Network (GNOME ▸ Система ▸ Администрирование ▸ Сеть)) позволяет использовать GUI для настройки сетевого интерфейса, сетевых устройств, серверов DNS, списка элементов сети и даже виртуальных частных сетей IPsec. Здесь же можно настроить динамические (DHCP, bootp) и стационарные IP-адреса. Есть даже возможность настраивать статичную сетевую маршрутизацию.

Однако в некоторых случаях работа сетевых интерфейсов может нарушиться: например, способы работы с сетевыми интерфейсами могут не поддерживаться GUI. Именно для подобных случаев в следующих разделах приведено описание способов работы с сетевыми интерфейсами из командной строки.

Карты сетевого интерфейса

Если сетевые устройства компьютера не позволяют сразу же после загрузки установить соединение с Интернетом, то для решения проблемы необходимо выполнить несколько последовательных операций.

1. Убедиться, что карта сетевого интерфейса (NIC) правильно установлена и сетевой кабель подключен (ISP CPE, перемычка и т. д.).
2. После проверки соединения убедиться, что нет сообщения об отсутствии соединения или несоответствии его дуплексной передаче.
3. Если ничего не помогает, необходимо заменить старый NIC на заведомо исправный, чтобы исключить возможность наличия аппаратных проблем.

Для проверки наличия сообщения и определения скорости и дуплексности предусмотрены две возможные команды: уже устаревшая mii-tool (программный пакет net-tools) и более новая ethtool (программный пакет ethtool). Если у вас не используется старый драйвер NIC, несовместимый с командой ethtool, то рекомендуется всегда использовать команду ethtool.

Для просмотра синтаксиса команды ethtool выполните следующую команду:

```
$ ethtool -h | less          Отображает параметры команды ethtool
```

Возвращаемая командой ethtool информация является встроенной в stderr помощью. Чтобы просмотреть эти сообщения с помощью less, мы перенаправили stderr в stdout.

Чтобы **отобразить настройки конкретной карты Ethernet**, добавьте к данной команде имя соответствующего интерфейса. Например, чтобы просмотреть информацию об eth0, нужно ввести:

```
$ sudo ethtool eth0          Отображает параметры NIC интерфейса eth0
Settings for eth0:
        Supported ports: [ TP MII ]
        Supported link modes: 10baseT/Half 10baseT/Full
                              100baseT/Half 100baseT/Full
        Supports auto-negotiation: Yes
        Advertised link modes: 10baseT/Half 10baseT/Full
                               100baseT/Half 100baseT/Full
        Advertised auto-negotiation: Yes
        Speed: 100Mb/s
        Duplex: Full
        Port: MII
        PHYAD: 1
        Transceiver: internal
        Auto-negotiation: on
        Supports Wake-on: g
        Wake-on: g
        Current message level: 0x00000007 (7)
        Link detected: yes
```

Для получения информации об интерфейсе Ethernet могут потребоваться привилегии суперпользователя — именно с этим связано использование в предыдущем примере команды sudo.

Чтобы **получить информацию о драйвере конкретной сетевой карты**, воспользуйтесь параметром -i:

```
$ sudo ethtool -i eth0        Выводит информацию о драйвере NIC
driver: e1000
version: 7.3.15-k2-NAPI
firmware-version: 0.5-7
bus-info: 0000:04:00.0
```

Чтобы **отобразить подробную статистику использования NIC**, необходимо использовать параметр -S:

```
$ sudo ethtool -S eth0        Отображает статистику NIC интерфейса eth0
NIC statistics:
   rx_packets: 1326384
   tx_packets: 773046
   rx_bytes: 1109944723
   tx_bytes: 432773480
   rx_errors: 5
   tx_errors: 2
   rx_dropped: 0
   tx_dropped: 0
   multicast: 0
   collisions: 0
   rx_length_errors: 0
   rx_over_errors: 0
   rx_crc_errors: 5
   rx_frame_errors: 0
   rx_fifo_errors: 0
   rx_missed_errors: 0
   tx_aborted_errors: 0
   tx_carrier_errors: 2
 ...
```

Команда ethtool может использоваться как для **изменения настроек NIC**, так и для их отображения. Для отключения автоматического согласования и фиксирования скорости NIC, равной 100 Мбит/сек при полном дуплексе, необходимо выполнить следующую команду:

```
$ sudo ethtool -s eth0 speed 100 duplex full autoneg off  Изменяет настройки NIC
```

Для отключения же автоматического согласования и фиксирования скорости, равной 10 Мбит/сек при полудуплексе, необходимо выполнить следующую команду:

```
$ sudo ethtool -s eth0 speed 10 duplex half autoneg off  Изменяет настройки NIC
```

Изменения, произведенные в настройках NIC, распространяются и на текущую сессию, однако после перезагрузки эти изменения сбрасываются. **Чтобы изменения действовали и после перезагрузки компьютера или сети**, необходимо создать исполняемый файл, который будет выполняться во время загрузки системы.

1. Выберите имя для файла (например, eth_options), а затем создайте файл в каталоге /etc/init.d:

   ```
   $ sudo vi /etc/init.d/eth_options
   ```

2. Вставьте в этот файл следующий текст:

```
#!/bin/sh
ETHTOOL="/usr/sbin/ethtool"
ETHTOOL_OPTS="speed 10 duplex half autoneg off"
DEV="eth0"
case "$1" in
start)
      echo -n "Setting $DEV options to $ETHTOOL_OPTS...";
      $ETHTOOL -s $DEV $ETHTOOL_OPTS;
      echo " done.";;
stop)
      ;;
esac
exit 0
```

3. Поместите специфические настройки в переменную ETHTOOL_OPTS, например:

```
ETHTOOL_OPTS="speed  10  duplex  half  autoneg  off"
```

Кроме того, измените переменную DEV, указывающую на первый интерфейс Ethernet – eth0.

4. Присвойте файлу свойства исполняемого:

```
$ sudo chmod +x /etc/init.d/eth_options
```

5. Создайте символьные ссылки для запуска файла на различных уровнях выполнения:

```
$ sudo update-rc.d eth_options defaults
Adding system startup for /etc/init.d/eth_options ...
  /etc/rc0.d/K20eth_options -> ../init.d/eth_options
  /etc/rc1.d/K20eth_options -> ../init.d/eth_options
  /etc/rc6.d/K20eth_options -> ../init.d/eth_options
  /etc/rc2.d/S20eth_options -> ../init.d/eth_options
  /etc/rc3.d/S20eth_options -> ../init.d/eth_options
  /etc/rc4.d/S20eth_options -> ../init.d/eth_options
  /etc/rc5.d/S20eth_options -> ../init.d/eth_options
```

Запустить созданный файл вы можете с помощью следующей команды:

```
$ sudo /etc/init.d/eth_options start
```

ПРИМЕЧАНИЕ

Подобные советы можно найти на сайте nixCraft (www.cyberciti.biz/tips).

Как было отмечено ранее, команда ethtool может **не работать со старыми картами NIC**. Если у вас именно такая ситуация, попробуйте воспользоваться командой mii-tool:

```
$ sudo mii-tool        Отображает согласованную скорость и статус
                       сообщения старой NIC
eth0: negotiated 100baseTx-FD flow-control, link ok
```

Данная команда была выполнена на том же компьютере, что и предыдущие: с автоматическим согласованием NIC на скорости 1000 Мбит/сек при полном

дуплексе. Команда mii-tool не может получить настройки скорости, поэтому мы рекомендуем использовать утилиту mii-tool только в качестве последнего средства, когда невозможно использовать ethtool.

Для **отображения с помощью mii-tool более подробной информации** предназначен параметр -v:

```
$ sudo mii-tool -v          Выдает подробное описание параметров NIC
eth0: negotiated 100baseTx-FD flow-control, link ok
  product info: vendor 00:50:43, model 12 rev 2
  basic mode:   autonegotiation enabled
  basic status: autonegotiation complete, link ok
  capabilities: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD
  advertising:  100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD flow-control
  link partner: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD flow-control
```

В данном примере показано, что оба модуля (100baseTx и 10baseT) поддерживают полудуплекс (HD) и полный дуплекс (FD). Чтобы **отключить автоматическое согласование и самостоятельно указать настройки**, нужно использовать параметр -F:

```
$ sudo mii-tool -F 10baseT-FD eth0       Устанавливает значения скорости
                                         и дуплекса равными 10baseT-FD
```

Чтобы позже **снова включить автоматическое согласование**, воспользуйтесь параметром -r:

```
$ sudo mii-tool -r eth0     Восстанавливает автоматическое согласование для NIC
restarting autonegotiation...
```

Команда mii-tool, в отличие от ethtool, не предоставляет возможности сохранять настройки, поэтому эту команду необходимо выполнять после каждой перезагрузки. Чтобы это сделать, добавьте ее в конец файла /etc/rc.local.

Команда netstat предоставляет еще один способ **отобразить статистику сетевого интерфейса**:

```
$ netstat -I          Отображает статистику сетевого интерфейса eth0
Kernel Interface table
Iface MTU Met RX-OK RX-ERR RX-DRP RX-OVR TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0 1500 0 1757208      6      0      0 996834     4      0      0 BMRU
```

Чтобы с помощью команды netstat **каждую секунду обновлять статистику сетевого интерфейса**, введите параметр -c:

```
$ netstat -ic         Обновляет статистику сетевого интерфейса ежесекундно
```

Объединив данную команду с командой watch, можно **получить более четкий, ориентированный на экран вывод** команды netstat:

```
$ watch netstat -i          Обновляет статистику сети (в режиме,
                            ориентированном на экран)
Every 2.0s: netstat -i                          Wed Aug 22 01:55:48 2007
Kernel Interface table
Iface MTU Met RX-OK RX-ERR RX-DRP RX-OVR TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0 1500 0 1757208      6      0      0 996834     4      0      0 BMRU
```

Как видно из отображенной в данном примере информации, статистика netstat обновляется каждые две секунды.

Управление подключениями к сети

Запуск и остановка сетевых интерфейсов ваших Ethernet-соединений к сетям LAN или к Интернету обычно происходит автоматически при загрузке и выключении Ubuntu соответственно. Однако для запуска или остановки сетевых интерфейсов можно использовать команды из /etc/init.d, а для определения автоматически загружаемой сети — из update-rc.d.

Команды ifconfig и ip могут также использоваться для настройки, активации и деактивации интерфейсов. Однако на Ubuntu и других дистрибутивах, основанных на Debian, команды из каталога /etc/init.d предоставляют более простые способы запуска и настройки сетевых интерфейсов. По этой причине в большинстве случаев для сбора информации об интерфейсах Ethernet и NIC предпочтительнее использовать команды ifconfig и ip (что будет показано позже в данном разделе).

Запуск и остановка Ethernet-подключений

Чаще всего соединения Ethernet запускаются вместе с Ubuntu, поскольку включен сетевой сервис при общих уровнях выполнения загрузки (уровни выполнения 3 и 5). Существует несколько отвечающих за это конфигурационных и исполняемых файлов, а также несколько простых команд, позволяющих пользователям управлять этим.

В случае с Ubuntu контрольные исполняемые и конфигурационные файлы хранятся в каталоге /etc/network/. Карты NIC настраиваются посредством редактирования файла /etc/network//interfaces, который выглядит следующим образом:

```
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet dhcp
auto eth1
iface eth1 inet dhcp
auto eth2
iface eth2 inet dhcp
auto ath0
iface ath0 inet dhcp
auto wlan0
iface wlan0 inet dhcp
```

Для **получения дополнительной информации об этом файле** введите следующее:

```
$ less /usr/share/doc/network-manager/README.Debian
```

Если вы изменили файл интерфейса, необходимо ввести следующую команду:

```
$ sudo /etc/dbus-1/event.d/25NetworkManager restart
```

Сценарий /etc/init.d/network запускает настроенные сетевые сценарии. Как и в других Linux-системах, вы можете запустить и остановить сетевую службу с помощью команды /etc/init.d/networking command.

Для того чтобы какие-либо изменения сетевых сценариев вступили в силу, вам необходимо **остановить, а затем запустить все NIC**. Для этого введите следующее:

```
$ sudo /etc/init.d/networking restart        Остановка и запуск
                                             сетевых интерфейсов
* Reconfiguring network interfaces...
There is already a pid file /var/run/dhclient.eth0.pid with pid 9242
killed old client process, removed PID file
Internet Systems Consortium DHCP Client V3.0.4
Copyright 2004-2006 Internet Systems Consortium. All rights reserved.
For info, please visit http://www.isc.org/sw/dhcp/

Listening on LPF/eth0/00:19:d1:5a:a9:e2
Sending on     LPF/eth0/00:19:d1:5a:a9:e2
Sending on     Socket/fallback
DHCPRELEASE on eth0 to 192.168.1.1 port 67
There is already a pid file /var/run/dhclient.eth0.pid with pid 134993416
Internet Systems Consortium DHCP Client V3.0.4
Copyright 2004-2006 Internet Systems Consortium.
...
[ OK ]
```

Вы можете наблюдать ошибки, связанные с дополнительными устройствами, указанными в конфигурации, но которые на самом деле отсутствуют в системе, как, например, оборудование для беспроводной связи. Можете игнорировать все ошибки, связанные с устройствами, которые вы не устанавливали.

Используйте параметр **остановки и запуска ваших сетевых интерфейсов**:

```
$ sudo /etc/init.d/networking stop      Остановка сетевых интерфейсов
$ sudo /etc/init.d/networking start     Запуск сетевых интерфейсов
```

Для **проверки состояния ваших сетевых интерфейсов** введите следующее:

```
$ ifconfig     Проверка состояния сетевых интерфейсов
eth0      Link encap:Ethernet HWaddr 00:19:D1:5A:A9:E2
          inet addr:192.168.1.106 Bcast:192.168.1.255 Mask:255.255.255.0
          inet6 addr: fe80::219:d1ff:fe5a:a9e2/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST   MTU:1492   Metric:1
          RX packets:14442 errors:0 dropped:0 overruns:0 frame:0
          TX packets:13080 errors:0 dropped:0 overruns:0 carrier:0
          collisions:434 txqueuelen:1000
          RX bytes:3732823 (3.5 MiB)   TX bytes:1142020 (1.0 MiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
```

```
          UP LOOPBACK RUNNING   MTU:16436   Metric:1
          RX packets:35 errors:0  dropped:0 overruns:0 frame:0
          TX packets:35 errors:0  dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:2121 (2.0 KiB)   TX bytes:2121 (2.0 KiB)
```

Если у вас есть несколько сетевых интерфейсов, возможно, вы захотите **отключить или запустить** один из них. Для этого используйте команды ifup и ifdown:

```
$ sudo ifdown eth0    Отключить сетевой интерфейс eth0
$ sudo ifup eth0      Включить сетевой интерфейс eth0
```

Существуют инструменты для просмотра информации о работающих сетевых интерфейсах и соответствующих им сетевых картах.

Просмотр информации об Ethernet-подключениях

Для просмотра адреса управления доступом к среде передачи данных (MAC-адреса) TCP/IP-подключений для вашей сетевой карты и IP-адреса используется команда ifconfig. Следующая командная строка **отображает информацию об адресе и статусе Ethernet-интерфейса eth0**:

```
$ ifconfig eth0
eth0      Link encap:Ethernet   HWaddr 00:D0:B7:79:A5:35
          inet addr:10.0.0.155   Bcast:10.0.0.255   Mask:255.255.255.0
          inet6 addr: fe80::2d0:b7ff:fe79:a535/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST   MTU:1500   Metric:1
          RX packets:1413382 errors:6 dropped:0 overruns:0 frame:6
          TX packets:834839 errors:4 dropped:0 overruns:0 carrier:4
          collisions:0 txqueuelen:1000
          RX bytes:1141608691 (1.0 GiB)   TX  bytes:470961026 (449.1 MiB)
```

В этом примере eth0 является первым Ethernet-интерфейсом на компьютере. MAC-адрес (HWaddr) сетевой карты в данном случае таков: 00:D0:B7:79:A5:35. IP-адрес eth0 – это 10.0.0.155, широковещательный адрес – 10.0.0.255, а маска подсети – 255.255.255.0. Остальная информация включает в себя количество переданных и полученных пакетов, а также возникшие проблемы (ошибки, утерянные пакеты и переполнение), связанные с сетевым интерфейсом.

Для **получения информации об активных и неактивных сетевых картах** используйте параметр -a.

```
$ ifconfig -a
```

Вместо команды ifconfig (и нескольких других команд, описанных в этой главе) можно использовать новую команду ip. Она создавалась для отображения информации о сетевых интерфейсах. Кроме того, она позволяет изменить настройки сетевых устройств, маршрутизации и IP-туннелей. В данном примере команда ip **применяется для отображения информации об интерфейсе eth0**:

```
$ ip addr show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
```

```
    link/ether 00:d0:b7:79:a5:35 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.155/24 brd 10.0.0.255 scope global eth0
    inet6 fe80::2d0:b7ff:fe79:a535/64 scope link
          valid_lft forever preferred_lft forever
```

Команда ip сокращает используемый синтаксис. Она работает подобно интерфейсу командной строки в Cisco IOS. Например, вместо того чтобы набирать ip addr show, можно использовать следующий синтаксис для **получения информации обо всех интерфейсах**:

```
$ ip a
```

Команда ip функционирует в нескольких компонентах сети, известных как объекты. Одним из этих объектов является addr; он позволяет ip настраивать сетевые адреса. Объекты команды ip будут рассмотрены ниже.

Чтобы увидеть, как применяется команда ip, используйте параметр помощи. Наряду с этим вы можете задать параметры для **получения информации об использовании объекта**:

```
$ ip help              Просмотр информации об использовании ip
Usage: ip [ OPTIONS ] OBJECT { COMMAND | help }
      ip [ -force ] [-batch filename
where  OBJECT := { link | addr | route | rule | neigh | ntable | tunnel|
                          maddr | mroute | monitor | xfrm }
     OPTIONS := { -V[ersion] | -s[tatistics] | -r[esolve] |
                         -f[amily] { inet | inet6 | ipx | dnet | link } |
                         -o[neline] | -t[imestamp] }
$ ip addr help         Просмотр помощи по обьекту addr
$ ip route help        Просмотр помощи по обьекту route
$ ip tunnel help       Просмотр помощи по обьекту tunnel
```

Если вы не знакомы с принципами работы масок подсети, то их понимание может оказаться достаточно сложным. Вы можете использовать ipcalc (из пакета программ ipcalc) для **расчета маски подсети хост-компьютера с помощью его CIDR IP-адреса** (CIDR – маршрутизация между доменами без разделения на классы):

```
$ ipcalc -bmn 192.168.1.100/27
Address:  192.168.1.100
Netmask:  255.255.255.224 = 27
Wildcard: 0.0.0.31
=>
Network:  192.168.1.96/27
HostMin:  192.168.1.97
HostMax:  192.168.1.126
Broadcast:      192.168.1.127
Hosts/Net:      30                        Class C, Private Internet
```

В показанном примере 255.255.255.224 – это маска подсети (которая указывает, какая часть IP-адреса представляет сеть, а какая – хост). Она была рассчитана с помощью значения /27 в конце IP-адреса 192.168.1.100.

Беспроводные соединения

Ранее установить беспроводное соединение в Linux было достаточно сложно. Основная причина заключалась в отсутствии драйверов с открытым кодом для большинства беспроводных сетевых карт, доступных на рынке. С последними выпусками Ubuntu ситуация улучшилась.

В процессе настройки беспроводного соединения мы рекомендуем пользоваться возможностями графического интерфейса (в частности, описанное в данной главе окно Network Configuration (Конфигурация сети) или Network Manager (Диспетчер сети)) для базовых настроек. Вам может понадобиться установить пакеты программ со средствами для работы с беспроводными устройствами, например wireless-tools и bcm43xx-fwcutter, которые доступны в интернет-репозиториях Ubuntu. Кроме того, вам могут понадобиться программно-аппаратные средства, которые можно найти в следующих пакетах программ: ipw2100, встроенные ipw2200 и zd1211.

Если вы не смогли настроить беспроводную LAN-карту с помощью окна Network Configuration (Конфигурация сети), то, возможно, вам помогут драйверы и утилиты, доступные на проектах Atheros (www.atheros.com), MadWifi (www.madwifi.org) или Ndiswrapper (ndiswrapper.sourceforge.net). Многие установочные пакеты программ с этих проектов доступны на стандартных интернет-репозиториях Ubuntu, описанных в гл. 2.

Если вам необходимо узнать, **какая именно беспроводная сетевая карта установлена в вашей системе**, выполните следующую команду:

```
$ lspci | grep -i wireless        Поиск беспроводных PCI-карт
01:09.0 Network controller: Broadcom Corporation BCM4306 802.11b/g
  Wireless LAN Controller (rev 03)
```

Допустим, ваша беспроводная сетевая карта установлена и работает. Существует несколько полезных команд в рамках установочного пакета с утилитами для работы с беспроводными устройствами, которые можно использовать для просмотра и изменения настроек беспроводных сетевых карт. В частности, команда iwconfig может помочь в работе с беспроводными интерфейсами локальной сети. В следующем примере показывается, **как сканировать сетевые интерфейсы на наличие поддерживаемых беспроводных сетевых карт**, после чего выдается список текущих настроек:

```
$ iwconfig
eth0      no wireless extensions.
eth1      IEEE 802.11-DS  ESSID:"" Nickname:"HERMES I"
          Mode:Managed Frequency:2.457 GHz  Access Point: Not-Associated
          Bit Rate:11 Mb/s   Tx-Power=15 dBm  Sensitivity:1/3
          Retry limit:4      RTS thr:off  Fragment thr:off
          Encryption key:off
          Power Management:off
```

Беспроводные интерфейсы могут называться wlanX или ethX, в зависимости от оборудования и используемого драйвера. Вы можете получить больше информации с помощью следующей команды:

```
$ ip link set eth1 up
$ iwconfig eth1
eth1      IEEE 802.11-DS  ESSID:""  Nickname:"HERMES I"
          Mode:Managed  Frequency:2.457 GHz  Access Point: None
          Bit Rate:11 Mb/s  Tx-Power=15 dBm  Sensitivity:1/3
          Retry limit:4  RTS thr:off  Fragment thr:off
          Encryption key:off
          Power Management:off
          Link Quality=0/92  Signal level=134/153  Noise level=134/153
          Rx invalid  nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0  Missed beacon:0
```

Показанные настройки можно изменять разными способами. Рассмотрим несколько вариантов **использования команды iwconfig для изменения настроек беспроводного интерфейса**. В следующих примерах мы работаем с беспроводным интерфейсом под названием wlan0. Данные операции могут поддерживаться в вашей системе, но могут и не поддерживаться. Это зависит от беспроводной сетевой карты и драйвера, которые вы используете.

```
$ sudo iwconfig wlan0 essid "MyWireless"  Присвоить essid
                                          значение MyWireless
$ sudo iwconfig wlan0 channel 3           Установить канал 3
$ sudo iwconfig wlan0 mode Ad-Hoc         Сменить режим Managed
                                          на режим Ad-Hoc
$ sudo iwconfig wlan0 ap any              Использовать любую доступную
                                          точку доступа
$ sudo iwconfig wlan0 sens -50            Установить чувствительность -50
$ sudo iwconfig wlan0 retry 20            Установить значение 20
                                          для повторной передачи MAC
$ sudo iwconfig wlan0 key 1234-5555-66    Установить ключ кодирования
                                          1234-5555-66
```

Параметр essid иногда называют сетевым именем или доменом. Используйте его как имя для вашей беспроводной сети. Установка канала позволяет вашей беспроводной локальной сети работать на определенном канале.

В режиме Ad-Hoc сеть состоит только из взаимосоединенных клиентов без центральной точки доступа. Установив определенный MAC-адрес точки доступа (ap) в режиме Managed/Infrastructure, вы можете дать команду сетевой карте подключаться к точке доступа с этим адресом или же установить параметр ap в состояние any — это разрешит подключение к любой точке доступа.

Если есть проблемы с производительностью, попробуйте установить положительное (это или процентное соотношение, или значение чувствительности, установленное продавцом) или отрицательное значение чувствительности. Если не удается провести повторную передачу, можно увеличить значение retry, чтобы ваша карта смогла посылать больше пакетов перед отказом.

Используйте параметр key для задания ключа кодирования. Вы можете вводить шестнадцатеричные цифры (XXXX-XXXX-XXXX-XXXX или XXXXXXXX). Добавив символ s: перед ключом, вы можете установить символ ASCII в качестве ключа (например, s:My927pwd).

Использование модемов коммутируемой линии передач

Хотя в наши дни высокоскоростные DSL-модемы и беспроводное сетевое оборудование широко распространены, все же иногда случается так, что телефонная линия и обычный модем остаются единственными инструментами для подключения к Интернету. Linux предоставляет как графические средства, так и средства командной строки для настройки и работы с модемами.

Как и другие сетевые подключения в Ubuntu, модемы коммутируемой линии передач могут быть настроены с помощью окна Network Configuration (Конфигурация сети). Большинство внешних модемов будут работать в Ubuntu без специальной настройки. То же самое можно сказать и про большинство внутренних PCI-модемов. Однако программные модемы (их иногда называют Win-модемы) часто не работают в среде Linux (хотя некоторые могут быть настроены с помощью специальных драйверов и поэтому относятся к Lin-модемам).

Вместо описания всевозможных ухищрений, с помощью которых можно заставить некоторые Win-модемы работать в Linux, мы рекомендуем вам приобрести модем, который подключается к последовательному порту, или аппаратный модем. Если же вы хотите настроить свой Win-модем самостоятельно, обратитесь на сайт www.linmodems.org.

Если у вас не получается настроить модем через окно Network Configuration (Конфигурация сети), можно использовать еще несколько команд. Для начала используйте команду `wvdialconf` для **поиска модемов, подключенных к последовательному порту, и создания файла конфигурации**:

```
$ sudo wvdialconf /etc/wvdial.conf        Сканирование последовательных портов,
                                          создание файла конфигурации
Scanning your serial ports for a modem.
ttyS0: ATQ0 V1 E1 -- OK
ttyS0: ATQ0 V1 E1 Z -- OK
...
```

В этом примере был обнаружен модем, подключенный к порту COM1 (последовательный порт `/dev/ttyS0`). Дальнейшая информация должна отобразить доступные скорости и различные поддерживаемые функциональные возможности.

Информация о конфигурации в этом случае записывается в файл `/etc/wvdial.conf`. Вот пример того, как он может выглядеть:

```
[Dialer Defaults] Modem = /dev/ttyS0
Baud = 115200
Init1 = ATZ
Init2 = ATQ0 V1 E1 S0=0 &C1 &D2 S11=55 +FCLASS=0
;Phone =
;Username =
;Password =
```

Откройте файл `wvdial.conf` в текстовом редакторе и удалите символ комментария (`;`) перед элементами списка `Phone`, `Username` и `Password`. Затем введите необхо-

димый телефонный номер для подключения к модемному пулу вашего провайдера. После этого добавьте имя пользователя и пароль подключения.

Для **использования этой конфигурации** вы можете применить команду wvdial:

```
$ sudo wvdial          Подключение к внешним службам по телефонной линии
                       и соединение с вашим провайдером
--> WvDial: Internet dialer version 1.54.0
--> Initializing modem.
--> Sending: ATZ
ATZ
OK
--> Modem initialized.
...
```

После установления соединения между двумя модемами между двумя точками создается протокол PPP. После этого вы наверняка сможете работать в Интернете.

Если модем все-таки не может посылать и принимать сигналы из Интернета, можно использовать еще несколько способов опроса последовательных портов вашего компьютера для установления неполадок. Первое, что необходимо сделать, — проверить, идет ли обмен информацией между устройством /dev/ttyS? и последовательным портом.

По умолчанию в системе Linux установлено четыре последовательных порта: COM1 (/dev/ttyS0), COM2 (/dev/ttyS1), COM3 (/dev/ttyS2) и COM4 (/dev/ttyS3). Для отображения их списка применяйте команду setserial (из установочного пакета setserial) с аргументом -g, как показано в примере:

```
$ setserial -g /dev/ttyS0 /dev/ttyS1 /dev/ttyS2 /dev/ttyS3
   Отобразить информацию о порте
/dev/ttyS0, UART: 16550A, Port: 0x03f8, IRQ: 4
/dev/ttyS1, UART: unknown, Port: 0x02f8, IRQ: 3
/dev/ttyS2, UART: unknown, Port: 0x03e8, IRQ: 4
/dev/ttyS3, UART: unknown, Port: 0x02e8, IRQ: 3
```

Для **более детальной информации о ваших последовательных портах** используйте параметр -a:

```
$ setserial -a /dev/ttyS0               Отобразить детальную информацию о порте
/dev/ttyS0, Line 0, UART: 16550A, Port: 0x03f8, IRQ: 4
   Baud_base: 115200, close_delay: 50, divisor: 0
   closing_wait: 3000
   Flags: spd_normal skip_test
$ setserial -ga /dev/ttyS0 /dev/ttyS1   Отобразить детальную информацию
                                        о нескольких портах
```

Команда setserial также может использоваться для перераспределения физических последовательных портов на логические устройства /dev/ttyS?. Если ваша система не работает на ядре 2.2 с платой последовательного доступа ISA, настроенной в режим переключателя, то вам не понадобится выполнять эту операцию. В Linux-системах с современным оборудованием порты COM1 и COM2 работают правильно, поэтому мы не будем рассматривать эти возможности.

Еще одной командой для работы с последовательными портами является stty. Для **просмотра текущих настроек порта COM1 (ttyS0)** проделайте следующее:

```
$ stty -F /dev/ttyS0 -a      Просмотр настроек tty для последовательного порта
speed 9600 baud; rows 0; columns 0; line = 0;
intr = ^C; quit = ^\; erase = ^?; kill = ^U; eof = ^D; eol = <undef>;
eol2 = <undef>; swtch = <undef>; start = ^Q; stop = ^S;
susp = ^Z; rprnt = ^R; werase = ^W; lnext = ^V; flush = ^O; min = 1; time = 0;
-parenb -parodd cs8 hupcl -cstopb cread clocal -crtscts
-ignbrk -brkint -ignpar -parmrk -inpck -istrip -inlcr
-igncr icrnl ixon -ixoff -iuclc -ixany -imaxbel -iutf8
opost -olcuc -ocrnl onlcr -onocr -onlret -ofill
-ofdel nl0 cr0 tab0 bs0 vt0 ff0 isig icanon iexten echo echoe echok
-echonl -noflsh -xcase -tostop -echoprt echoctl echoke
```

Устройство автоматического набора номера обычно меняет эти настройки на нужные, однако вы также можете использовать команду stty для их изменения.

Вы можете обращаться напрямую к модему или другому последовательному устройству с помощью команды minicom (из установочного пакета minicom). К тому же это может быть полезно при устранении неполадок путем послания модему AT-команд с помощью minicom. При первом запуске пакета используйте параметр -s для входа в режим настройки:

```
$ minicom -s           Создание настроек для вашего модема
+-----[configuration]-------+
| Filenames and paths       |
| File transfer protocols   |
| Serial port setup         |
| Modem and dialing         |
| Screen and keyboard       |
| Save setup as dfl         |
| Save setup as..           |
| Exit                      |
| Exit from Minicom         |
+---------------------------+
```

Забудем на некоторое время о модемах и предположим, что с помощью COM1 вы хотите подключиться к устройству Cisco на скорости 9600 бод. С помощью клавиш управления курсором стрелок на клавиатуре выберите **Serial port setup** (Установка последовательного порта) и нажмите клавишу **Enter**. Нажмите клавишу **A**, чтобы отредактировать устройство для последовательной передачи данных, и измените его на /dev/ttyS0. Далее нажмите клавишу **E** для доступа к настройкам порта и, когда появится экран **Comm Parameters** (Параметры связи), нажмите **E** для выбора значения **9600**. Для отключения аппаратного управления потоками нажмите клавишу **F**. После этого нажмите **Enter** для возврата в меню конфигурации.

Для изменения параметров модема выберите команду **Modem and dialing** (Модем и дозвон). Затем очистите строки **init**, **reset**, **connect** и **hangup** (они несовместимы с настраиваемым устройством Cisco). После этого в экране установки выберите **Save Setup as df1** (Сохранить конфигурацию как df1) (по умолчанию) и выберите **Exit** (здесь: выход; но не выход из самой Minicom).

Теперь вы находитесь в терминале minicom. Для детальной информации по использованию minicom нажмите сочетание клавиш **Ctrl+A**, а затем **Z** для отображения помощи. Когда закончите работу с minicom, нажмите сочетание **Ctrl+A**, а затем **X** для выхода.

ВНИМАНИЕ

Не запускайте minicom в окне с конфигурацией клавиш, установленной по умолчанию, иначе сочетание Ctrl+A будет перехвачено окном! Если вы сделали это по ошибке, перейдите в другое окно и наберите killall minicom.

Установка разрешения имен

Поскольку люди привыкли обращаться к различным вещам по их именам, а IP-адрес состоит из цифр, то такие сети, как Интернет, использующие протокол TCP/IP, полагаются на DNS для преобразования имен хост-узлов в IP-адреса. Ubuntu предоставляет возможность пользоваться средствами для поиска информации, относящейся к разрешению DNS-имен.

Во время первой установки Ubuntu вы либо установили DNS-серверы для работы с разрешением имен, либо позволили им быть назначенными автоматически с помощью сервера DHCP. Эта информация сохраняется в файле `/etc/resolv.conf`, который выглядит примерно так:

```
nameserver 11.22.33.44
nameserver 22.33.44.55
```

Цифры, показанные выше, заменяются настоящими IP-адресами компьютеров, которые являются серверами DNS. Когда есть возможность подключиться к рабочим DNS-серверам, вы можете использовать команды, которые применяются для опроса этих серверов и поиска хост-компьютеров.

Команда `dig` (которая используется вместо исключенной команды `nslookup`) применяется для получения информации от DNS-сервера. Команда `host` используется для получения адресной информации имени хоста или доменного имени.

Для **поиска определенного имени в вашем DNS-сервере** (www.turbosphere.com в следующем примере) используйте команду `dig` следующим образом:

```
$ dig www.turbosphere.com          Поиск по DNS-серверам,
                                   указанным в /etc/resolv.conf
```

Вместо использования приписанного системе сервера вы можете **сделать запрос к заданному блоку преобразования имен**. Следующий пример проводит опрос DNS-сервера по адресу 4.2.2.1:

```
$ dig www.turbosphere.com @4.2.2.1
```

С помощью команды `dig` также можно сделать **запрос об определенном типе записи**:

```
$ dig turbosphere.com mx    Запрос об устройстве работы с электронной почтой
$ dig turbosphere.com ns    Запрос о низкоуровневых серверах имен
```

Используйте параметр +trace, чтобы **отследить рекурсивный запрос** от высокоуровневого DNS-сервера к низкоуровневым серверам:

```
$ dig +trace www.turbosphere.com   Рекурсивное отслеживание DNS-серверов
```

Используйте параметр +short, если хотите **просмотреть IP-адрес хост-компьютера**:

```
$ dig +short www.turbosphere.com   Отображает только пару имя/IP-адрес
turbosphere.com.
66.113.99.70
```

Вы можете использовать команду dig **для «обратного» запроса, чтобы найти информацию о DNS по IP-адресу**:

```
$ dig -x 66.113.99.70 Получение информации о DNS по IP-адресу
```

Вы можете также использовать команду host **для «обратного» DNS-запроса**:

```
$ host 66.113.99.70
70.99.133.66.in-addr.arpa domain name pointer boost.turbosphere.com.
```

Применяйте команды hostname и dnsdomainname, чтобы **получить информацию об имени хоста локального компьютера**:

```
$ hostname             Просмотр полного DNS-имени локального компьютера
boost.turbosphere.com
```

Вы также можете использовать команду hostname **для установки временного имени хоста** (до следующей перезагрузки). Вот пример данной операции:

```
$ sudo hostname server1.example.com      Установить локальное имя хоста
```

Изменение имени хоста на работающем компьютере может отрицательно сказаться на некоторых работающих демонах. Мы рекомендуем **устанавливать локальное имя хоста во время каждого запуска системы**. Отредактируйте первую строку файла /etc/hostname. Например:

```
server1.example.com
```

Устранение неполадок в работе сети

Устранение неполадок, связанных с работой сети, обычно проводят начиная с нижнего уровня. Как уже говорилось в этой главе, первое, что необходимо сделать, — проверить, что все физические компоненты сети (кабели, сетевые карты и т. д.) правильно подключены и находятся в рабочем состоянии. Далее проверьте, что соединения между физическими узлами работают. После этого следует проверить соединение с определенным хостом. Для этого существует множество способов.

Проверка соединения с элементом сети

Если вы знаете, что соединение есть и отсутствует дуплексное рассогласование, то следующим шагом должна быть посылка запросов установленному по умолчанию

шлюзу. Вы либо сами настраивали шлюз, используемый по умолчанию в файле /etc/network/interfaces, либо позволили системе установить его с помощью службы DHCP (Протокол динамической настройки конфигурации хост-машины). Для того чтобы **узнать, какой шлюз используется по умолчанию**, используйте команду ip, как показано на примере:

```
$ ip route
10.0.0.0/24 dev eth0  proto kernel  scope link  src 10.0.0.155
169.254.0.0/16 dev eth0  scope link
default via 10.0.0.1 dev eth0
```

Шлюзом, используемым по умолчанию, в данном примере является 10.0.0.1. Чтобы убедиться в наличии IP-соединения с этим шлюзом, используйте команду ping, передавая адреса этому шлюзу, как показано в следующем примере:

```
$ ping 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=0.382 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.313 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.360 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=1.43 ms
--- 10.0.0.1 ping statistics ---
packets transmitted, 4 received, 0% packet loss, time 2999ms
rtt min/avg/max/mdev = 0.313/0.621/1.432/0.469 ms
```

Команда ping по умолчанию продолжает свою работу до тех пор, пока вы не нажмете сочетание клавиш Ctrl+C. Ниже перечислены другие параметры этой команды:

```
$ ping -a 10.0.0.1          Добавить звуковой эффект по мере пингования
$ ping -c 4 10.0.0.1        Пропинговать 4 раза и выйти (используется
                            по умолчанию в Windows)
$ ping -q -c 5 10.0.0.1     Отобразить сводку работы программы ping
                            (работает лучше с параметром -c)
$ sudo ping -f 10.0.0.1     Посылать большое количество пакетов (флудить)
$ ping -i 3 10.0.0.1        Отправлять пакеты с 3-секундным интервалом
$ sudo ping -I eth0 10.0.0.1 Сменить источник на eth0 (применяется
                            при наличии нескольких сетевых карт)
PING 10.0.0.1 (10.0.0.1) from 10.0.0.155 eth0: 56(84) bytes of data.
$ sudo ping -I 10.0.0.155 10.0.0.1 Сменить источник на 10.0.0.155
PING 10.0.0.1 (10.0.0.1) from 10.0.0.155 : 56(84) bytes of data.
$ ping -s 1500 10.0.0.1     Установить размер пакетов 1500 байт
PING 10.0.0.1 (10.0.0.1) 1500(1528) bytes of data.
```

Будьте внимательны при использовании параметра -f. По умолчанию команда ping посылает маленькие пакеты (56 байт). Большие пакеты (например, размером 1500 байт) хороши для обнаружения неисправных карт и соединений.

Проверка протокола разрешения адресов

Если вы не можете получить ответ шлюза на пинг, то могут быть проблемы на MAC-уровне сети Ethernet. Можно использовать протокол разрешения адресов

(ARP) для поиска информации на MAC-уровне. Чтобы просмотреть и проверить содержимое ARP, используйте смежные команды arp и ip. В этом примере показан список компьютеров по именам хостов, находящийся в кэше протокола разрешения адресов, который выводится с помощью команды arp:

```
$ arp -v                    Отобразить записи в кэше ARP по имени
Address        HWtype HWaddress                Flags Mask     Iface
Ritchie        ether  00:10:5A:AB:F6:A7  C                  eth0
Einstein  ether 00:0B:6A:02:EC:98  C                  eth0
Entries: 1  Skipped: 0  Found: 1
```

Этот пример показывает, как отобразить имена всех компьютеров, о которых существуют записи в локальном кэше протокола разрешения адресов. Кроме того, выводятся информация о соответствующем этим компьютерам оборудовании и аппаратные адреса (MAC-адреса) сетевых карт. Вы можете **отключить разрешение имен и увидеть IP-адреса компьютеров**:

```
$ arp -vn                   Отобразить записи в кэше ARP по IP-адресу
Address        HWtype       HWaddress                Flags Mask      Iface
10.0.0.1  ether        00:10:5A:AB:F6:A7  C                  eth0
10.0.0.50 ether        00:0B:6A:02:EC:98  C                  eth0
Entries: 1  Skipped: 0  Found: 1
```

Чтобы **удалить запись из кэша ARP**, используйте параметр -d:

```
$ sudo arp -d 10.0.0.50        Удалить адрес 10.0.0.50 из кэша ARP
```

Вместо того чтобы позволять протоколу разрешения имен динамически узнавать о других системах, вы можете **добавить статичные записи в кэш ARP** с помощью параметра -s:

```
$ sudo arp -s 10.0.0.51 00:0B:6A:02:EC:95       Добавить IP- и MAC-адреса в ARP
```

Для того чтобы выполнять те же действия, что и с помощью команды arp, посредством команды ip, используйте соседний объект (обратите внимание, что соседние объекты nei и n являются взаимозаменяемыми):

```
$ ip neighbor
10.0.0.1 dev eth0 lladdr 00:10:5a:ab:f6:a7 DELAY
10.0.0.50 dev eth0 lladdr 00:0b:6a:02:ec:98 REACHABLE
# ip nei del 10.0.0.50 dev eth0
# ip n add 10.0.0.51 lladdr 00:0B:6A:02:EC:95 dev eth0
```

Команда arping необходима **для создания запроса к подсети, проверяющего, используется ли уже такой IP-адрес**, и для нахождения MAC-адреса устройства, которое работает с этим IP. Команда ifup применяет arping, чтобы избежать конфликтов во время настройки Ethernet-карты. Рассмотрим соответствующие примеры:

```
$ arping 10.0.0.50                    Опросить подсеть, чтобы установить,
                                      используется ли адрес 10.0.0.50
ARPING 10.0.0.50 from 10.0.0.195 eth0
Unicast reply from 10.0.0.50 [00:0B:6A:02:EC:98]  0.694ms
Unicast reply from 10.0.0.50 [00:0B:6A:02:EC:98]  0.683ms
$ sudo arping -I eth0 10.0.0.50       Установить опрашиваемый интерфейс
```

Команда arping, подобно ping, продолжает посылать запросы, пока не будет остановлена нажатием сочетания клавиш **Ctrl+C**. Необходимо узнать, подает ли объект какие-либо признаки жизни, поэтому вы можете использовать одну из представленных команд:

```
$ arping -f 10.0.0.50       Опросить 10.0.0.50 и остановиться
                            после получения первого ответа
$ arping -c 2 10.0.0.51     Опросить 10.0.0.50 и остановиться после двух раз
```

Отслеживание маршрутов к хостам

Даже после того, как вы узнали, что можете пинговать шлюз и получать ответ от машин, находящихся вне вашей сети, у вас еще могут оставаться проблемы при соединении с определенным хост-компьютером или сетью. Если сложилась подобная ситуация, то можно использовать команду traceroute (из установочного пакета traceroute), **чтобы отыскать корень проблемы**:

```
$ traceroute boost.turbosphere.com  Следовать маршруту к определенному хост-компьютеру
traceroute to boost.turbosphere.com
   (66.113.99.70),30 hops max,40 byte packets
1  10.0.0.1 (10.0.0.1)  0.281 ms  0.289 ms 0.237 ms
2  tl-03.hbci.com (64.211.114.1) 6.213 ms 6.189 ms 6.083 ms
3  172.17.2.153 (172.17.2.153) 14.070 ms 14.025 ms 13.974 ms
4  so-0-3-2.ar2.MIN1.gblx.net (208.48.1.117) 19.076 ms 19.053 ms 19.004 ms
5  so1-0-0-2488M.ar4.SEA1.gblx.net(67.17.71.210)94.697 ms 94.668 ms 94.612ms
6  64.215.31.114 (64.215.31.114) 99.643 ms 101.647 ms 101.577 ms
7  dr02-v109.tac.opticfusion.net(209.147.112.50)262.301ms 233.316ms 233.153 ms
8  dr01-v100.tac.opticfusion.net (66.113.96.1) 99.313 ms 99.401 ms 99.353 ms
9  boost.turbosphere.com (66.113.99.70) 99.251 ms 96.215 ms 100.220 ms
```

Как вы видите, самый длинный переход находится между 4 (Global Crossing в Миннеаполисе) и 5 (группа компаний в Сиэтле). Этот интервал на самом деле не является основой проблемы; он отражает расстояние между двумя переходами. Иногда последние переходы выглядят так:

```
28 * * *
29 * * *
30 * * *
```

Строки со звездочками (*) могут означать, что сетевые устройства защиты (брандмауэр) блокируют трафик к этому объекту. Однако если несколько звездочек стоят перед адресом назначения, то они говорят о сильных перегрузках или отказе оборудования, что свидетельствует о причине проблемы.

Команда traceroute по умолчанию использует UDP-пакеты (UDP — протокол передачи пользовательских датаграмм), которые показывают более реалистичную картину, чем ICMP (ICMP — протокол управления сообщениями в сети). Это происходит из-за того, что некоторые интернет-узлы имеют низкий приоритет ICMP-трафика. Если вы все же хотите **использовать ICMP-пакеты при отслеживании**, попробуйте применить следующую команду:

```
$ traceroute -I boost.turbosphere.com    Использование ICMP-пакетов
                                         для отслеживания маршрутов
```

Команда traceroute по умолчанию соединяется с портом 80. Вы можете **использовать другой порт** посредством параметра -p:

```
$ traceroute -p 25 boost.turbosphere.com Проводить отслеживание
                                         через порт 25
```

Можно **просматривать IP-адреса вместо хост-имен**, отключив разрешение имен точек перехода:

```
$ traceroute -n boost.turbosphere.com    Отключить разрешение имен
                                         при отслеживании
```

Альтернативой traceroute является команда tracepath, которая использует UDP для отслеживания:

```
$ tracepath boost.turbosphere.com        Использовать UDP
                                         для отслеживания маршрутов
```

Команда route была лучшим средством для работы с таблицей маршрутизации ядра. Однако она понемногу заменяется командой ip route. Большинство сетевых сценариев Ubuntu полагается на ip route. Но не повредит, если вы ознакомитесь с обеими командами, так как до сих пор route используют довольно часто.

Вы можете применять старую команду route **для отображения вашей локальной таблицы маршрутизации**. Далее следуют два примера применения команды route — с разрешением имен DNS и без него:

```
$ route              Отобразить информацию о локальной таблице маршрутизации
Kernel IP routing table
Destination     Gateway            Genmask      Flags Metric  Ref  Use  Iface
10.0.0.0        *             255.255.255.0     U         0    0    0    eth0
Default         ritchie             0.0.0.0     UG        0    0    0    eth0
$ route -n           Отобразить таблицу маршрутизации без DNS-преобразований
Kernel IP routing table
Destination     Gateway            Genmask      Flags Metric  Ref  Use  Iface
10.0.0.0        *             255.255.255.0     U         0    0    0    eth0
0.0.0.0         10.0.0.1            0.0.0.0     UG        0    0    0    eth0
```

Можно **добавить шлюз, применяемый по умолчанию**, с помощью параметра gw:

```
$ sudo route add default gw 10.0.0.2    Установить 10.0.0.2 как шлюз,
                                        используемый по умолчанию
```

Вы можете **добавить новый маршрут в вашу сеть**, выбрав интерфейс (eth0) или IP-адрес шлюза (например, gw 10.0.0.100):

```
$ sudo route add -net 192.168.0.0 netmask 255.255.255.0 eth0
$ sudo route add -net 192.168.0.0 netmask 255.255.255.0 gw 10.0.0.100
```

С помощью параметра del можно **удалить маршрут**:

```
$ sudo route del -net 192.168.0.0 netmask 255.255.255.0     Удалить маршрут
```

Используя более новую команду ip, вы можете выполнять те же действия, которые были только что проделаны благодаря команде route. Существует три разных способа показа аналогичной базовой информации о маршрутизации:

```
$ ip route show         Показать основную информацию о маршрутизации (пример #1)
10.0.0.0/24 dev eth0 proto kernel scope link src 10.0.0.195
169.254.0.0/16 dev eth0 scope link default via 10.0.0.1 dev eth0
$ ip route              Показать основную информацию о маршрутизации (пример #2)
$ ip r                  Показать основную информацию о маршрутизации (пример #3)
```

Вот несколько примеров, демонстрирующих, как **добавлять и удалять маршруты** с помощью ip:

```
$ sudo ip r add 192.168.0.0/24 via 10.0.0.100 dev eth0  Добавить маршрут
                                                         в интерфейс
$ sudo ip r add 192.168.0.0/24 via 10.0.0.100           Добавить маршрут
                                                         без использования
                                                         интерфейса
$ sudo ip r del 192.168.0.0/24                          Удалить маршрут
```

Чтобы **сделать новый маршрут постоянным**, отредактируйте файл /etc/network/interfaces и поместите в него информацию о новом маршруте. К примеру, чтобы добавить маршрут, добавленный выше с помощью команды ip, поместите в этот файл следующие данные:

```
iface eth0 inet static
address 192.168.0.0
netmask 255.255.255.0
gateway 10.0.0.100
```

Просмотр соединений и статистики

Перечисленные выше средства по устранению неполадок в основном относятся к сетевому уровню (уровень 3). Для **вывода информации о пакетах, пересылаемых между протоколами транспортного уровня (TCP и UDP) и ICMP**, воспользуйтесь командой netstat:

Вы можете **просмотреть список всех TCP-подключений** и увидеть, какой процесс управляет соединением:

```
$ netstat -s | less        Показать сводку действий TCP, ICMP и UDP
$ sudo netstat -tanp Просмотреть активные TCP-подключения
Active Internet connections (servers and established)
Proto    Recv-QSend-QLocal Address       Foreign Address State  PID/Program name
tcp      0          0      127.0.0.1:631       0.0.0.0:*    LISTEN 2039/cupsd
tcp      0          0      127.0.0.1:25        0.0.0.0:*    LISTEN 2088/sendmail
```

Просмотреть активные UDP-подключения можно следующим образом:

```
$ sudo netstat -uanp Просмотреть активные UDP-подключения
Active Internet connections (servers and established)
Proto    Recv-QSend-QLocal Address       Foreign Address State  PID/Program name
udp      0          0      0.0.0.0:631         0.0.0.0:*           2039/cupsd
udp      0          0      192.168.122.1:123   0.0.0.0:*           2067/ntpd
```

Обратите внимание на слово listen — оно поможет **ограничить выводимые посредством команды netstat результаты до информации о демонах, связанных с портом TCP**. Например:

```
$ sudo netstat -tanp | grep -i listen    Просмотр демонов, прослушивающих порт
```

Данная команда является отличным способом разрешения конфликтов, связанных с использованием портов демонами.

Полезные утилиты для работы с сетью

Если вы хотите увидеть **информацию о пакетах, пересылаемых в системе**, используйте команду tcpdump. У нее есть много продвинутых функциональных возможностей, многие из которых завязаны на фильтрации и возможности найти «иголку в стоге пакетов с информацией». Если вы запустите команду tcpdump на удаленном компьютере, ваш экран наполнится ssh-трафиком между вашим клиентом и удаленной машиной. Чтобы слегка ознакомиться с tcpdump без необходимости изучать, как именно работает ее система фильтрации, выполните следующую команду:

```
$ sudo tcpdump | grep -v ssh    Поиск пакетов, исключая пакеты, связанные с ssh
```

Если же вы хотите заглянуть глубже, а именно до трафика на уровне пакетов, используйте команду wireshark (ранее известную как ethereal). Проинсталлируйте установочный пакет wireshark. На удаленном компьютере вы можете запускать его с параметром X при ssh. Wireshark является мощным анализатором пакетов, который конкурирует с лучшими коммерческими утилитами.

Для **исследования сетей и удаленных машин, а также для просмотра предлагаемых ими служб** используйте команду nmap. Эта команда (из установочного пакета nmap) является общепринятым сканером портов. Его даже показывали в фильме «Матрица: Перезагрузка»! Убедитесь, что вы имеете право на сканирование целевых систем или сетей. Команда nmap как часть установочного пакета nmap может быть запущена от имени обычного пользователя, однако для нескольких типов сканирования необходимы root-привилегии.

Рассмотрим, как с ее помощью **провести простое сканирование хостов**:

```
$ sudo nmap 10.0.0.1  Сканировать порты компьютера с адресом 10.0.0.1
```

Для **получения максимального количества информации** используйте параметр -vv:

```
$ sudo nmap -vv 10.0.0.1    Отображение максимального количества
                            информации в результатах nmap
```

Используйте сетевой адрес в качестве аргумента nmap, чтобы **провести сканирование всей сети**. В следующем примере мы добавили параметр -sP, чтобы провести простой пинг-обзор:

```
$ sudo nmap -vv -sP 10.0.0.0/24    Сканирует хост-компьютеры во всей сети
```

Вы можете получать от пакета nmap только необходимую информацию. В следующем примере параметр -P0 не позволяет использовать пингование (это отлично подходит для сканирования компьютеров, не отвечающих на пингование).

Параметр -O выводит информацию об операционных системах, установленных на сканируемых машинах. Параметр -p 100-200 указывает на то, что необходимо сканировать порты, начиная с сотого и заканчивая двухсотым:

```
$ sudo nmap -vv -P0 -O -p 100-200 10.0.0.1      Без пинга, информация об ОС,
                                                порты 100-200
```

У команды nmap есть множество других сложных и продвинутых параметров. Для получения более подробной информации обращайтесь к руководству по nmap.

Резюме

Почти каждый аспект сетевых подключений в Ubuntu может быть настроен, проверен и отслежен посредством командной строки. Вы можете просматривать и изменять настройки сетевых карт с помощью команд ethtool и mii-tool. Кроме того, есть возможность просмотреть сетевую статистику через netstat.

Простыми командами для запуска и остановки вашей сети являются service, chkconfig, ifup и ifdown. Когда соединение уже установлено, можно просматривать его статистику, используя команды ifconfig и ip.

Наряду с проводными Ethernet-картами, Linux поддерживает другие сетевые устройства, такие как беспроводные LAN-карты и модемы коммутируемой линии передач. Используйте команду iwconfig для работы с беспроводными интерфейсами, а команды wvdialconf и minicom — для настройки модемов.

Чтобы проверить разрешение имен DNS, применяйте команды dig, host и hostname. В команды для проверки соединений и маршрутов входят ping, arp, traceroute и ip.

12 Подключение к сетевым ресурсам

За время, потраченное на установку графического FTP-клиента, с удаленного сервера можно скачать десятки файлов, если использовать утилиты командной строки.

Команды для передачи файлов, поиска и просмотра информации в Сети, для доступа к папкам и чтения почты являются быстрыми и эффективными даже при наличии в системе графической оболочки. При отсутствии GUI они вообще становятся жизненно важными.

В этой главе описываются основы работы с командами для доступа к различным ресурсам (файлам, электронной почте, общим папкам и онлайн-чатам) посредством Сети.

Запуск программ для просмотра информации в Сети

С помощью текстовых браузеров можно быстро проверить, работает ли веб-сервер, либо получить от него необходимую информацию, когда не получается использовать графический интерфейс. Браузер lynx, некогда бывший популярным, в большинстве Linux-систем был вытеснен веб-обозревателем links, который позже заменили браузером elinks (на сегодняшний день при наборе команды `links` запускается браузер elinks). Для использования браузером командной строки необходимо установить одну из этих программ. Имена установочных пакетов совпадают с именами команд lynx, links и elinks соответственно. В большинстве случаев устанавливается пакет программ elinks.

Браузер elinks работает в окне терминала. Elinks может работать с такими основными функциями и содержимым HTML, как таблицы, фреймы, вкладки, cookie-файлы, журнал посещений, MIME-типы и простые каскадные таблицы стилей; однако он не позволяет просматривать изображения. Вы даже можете использовать мышь для перехода по ссылкам и работы с меню.

Если ваш терминал поддерживает разные цвета, то и в elinks эта возможность будет реализована. Это позволит вам с легкостью находить ссылки и заголовки в тексте (цвета могут не поддерживаться в рамках оконной сессии). Вот несколько примеров применения команды `elinks`:

```
$ elinks                              Запрашивает имя файла или URL
$ elinks www.handsonhistory.com       Открывает запрошенный файл или URL
```

Если у вас есть мышь, то щелкните ее кнопкой на верхней части терминала для отображения меню. Выберите необходимые пункты, а затем — ссылку. В табл. 12.1 показаны клавиши для навигации в программе elinks.

Таблица 12.1. Клавиши управления в программе elinks

Клавиши	Описание	Клавиши	Описание
Esc (или F9/F8)	Отобразить и скрыть меню (далее используйте клавиши для управления курсором или мышь для навигации по меню)	=	Просмотреть информацию о странице
Стрелка вниз	Перейти к следующей ссылке или редактируемому полю на странице	Ctrl+R	Перезагрузить страницу
Стрелка вверх	Перейти к предыдущей ссылке или редактируемому полю на странице	A	Добавить текущую страницу в закладки
Стрелка вправо или Enter	Перейти по выделенной ссылке. Ввести текст в выделенное поле формы	T	Открыть новую вкладку браузера
Стрелка влево	Перейти на предыдущую страницу	>	Перейти к следующей вкладке
/	Искать далее	<	Перейти к предыдущей вкладке
?	Искать в обратном направлении	C	Закрыть текущую вкладку
N	Продолжить поиск	D	Скачать файл по текущей ссылке
Shift+N	Найти предыдущее	Shift+D	Просмотреть закачки
Page Up	Пролистать одну страницу вверх	Shift+A	Добавить текущую ссылку в закладки
Page Down	Пролистать одну страницу вниз	S	Просмотреть закладки
G	Перейти по ссылке	V	Просмотреть текущее изображение
Q или Ctrl+C	Выйти из elinks	H	Открыть общий журнал посещений

Вы можете добавлять основные параметры elinks в файл `/etc/elinks.conf`. Персональные настройки пользователей хранятся в каждом каталоге `$HOME/.elinks`. Наберите `man elinkskeys`, чтобы увидеть возможные настройки.

Передача файлов

Команды для скачивания файлов с удаленных серверов в Linux-системах (HTTP, HTTPS, FTP или SSH) являются очень мощными. Вы можете выбрать для работы определенную команду, ориентируясь на наличие в ней нужных параметров. Например, вам может понадобиться провести закачку через шифрованное соединение,

продолжить прерванную закачку либо выполнить рекурсивную закачку. В данном разделе объясняются принципы использования команд wget, ftp, lftp, scp и scftp.

Закачка файлов с помощью команды wget

Иногда необходимо **скачать файл с удаленного сервера**, используя только командную строку. К примеру, вы нашли ссылку на пакет программ RMP, но она несколько раз перенаправляет на другие страницы, что не позволяет установить программу rpm напрямую с HTTP. Кроме того, возможно, вы захотите написать сценарий для автоматической закачки файла каждую ночь (например, файла журнала).

Команда wget позволяет скачивать файлы с FTP- и веб-серверов (HTTP и HTTPS). Если сервер не требует авторизации, то к команде wget можно просто добавить путь к искомому файлу:

```
$ wget https://help.ubuntu.com/7.04/common/img/headerlogo.png
```

Если же, к примеру, **FTP-сервер требует имя пользователя и пароль**, то можно ввести эту информацию в команду wget следующим образом:

```
$ wget ftp://user:password@ftp.example.com/path/to/file
$ wget --user=user --password=password ftp://ftp.example.com/path/to/file
```

Например:

```
$ wget ftp://chris:mykuulpwd@ftp.linuxtoys.net/home/chris/image.jpg
$ wget --user=chris --password=mykuulpwd \
ftp://ftp.linuxtoys.net/home/chris/image.jpg
```

Вы можете использовать команду wget следующим образом для **закачки одной веб-страницы**:

```
$ wget http://www.wiley.com          Закачать только веб-страницу
```

Однако если вы откроете сохраненный файл index.html, то увидите множество неработающих ссылок. Для того чтобы скачать все изображения и другие элементы, необходимые для правильного отображения страницы, используйте параметр -p:

```
$ wget -p http://www.wiley.com          Закачать веб-страницу и другие элементы
```

Но если вы теперь откроете index.html в браузере, то, возможно, увидите, что ссылки все еще остаются неисправными, хотя все рисунки были загружены. Это случается из-за того, что ссылки надо исправлять — они должны указывать на локальные файлы. В таком случае введите следующее:

```
$ wget -pk http://www.wiley.com     Закачать страницы с использованием
                                    локальных имен файлов
```

Если вы хотите, чтобы команда wget сохраняла исходные файлы, а также конвертировала их, наберите следующее:

```
$ wget -pkK http://www.wiley.com    Переименовать в локальные имена,
                                    сохранять исходные файлы
```

Иногда скачиваемые HTML-файлы имеют разрешение не HTML, а ASP или CGI. Это может привести к тому, что браузер не сможет открыть локальную копию файла. С помощью команды wget и параметра -E вы можете присоединить .html к названиям данных файлов:

```
$ wget -E http://www.aspexamples.com      Присоединить .html к названиям
                                          загруженных файлов
```

Команда wget может рекурсивно продублировать целый сайт. Параметр -m полностью копирует файлы и директории всей файловой структуры сервера и, помимо этого, отмечает время и сохраняет перечни файлов FTP-каталога.

ВНИМАНИЕ

Будьте осторожны с этим, так как такой процесс может занять много времени и места.

```
$ wget -m http://www.linuxtoys.net
```

Следующая строка команды содержит некоторые описанные выше параметры и **создает наиболее практичную копию сайта**:

```
$ wget -mEkK http://www.linuxtoys.net
```

Если у вас бывали случаи отключения от Сети во время закачки больших файлов (например, образа CD или DVD), то параметр -c команды wget может оказаться весьма полезным. Используя wget -c, вы предписываете программе **продолжить незавершенную закачку с того места, где она прервалась**. Например:

```
$ wget http://example.com/DVD.iso         Начать загрузку большого файла
   ...
95%[========= ] 685,251,583 55K/s         Закачка остановлена
                                          и является незавершенной
$ wget -c  http://example.com/DVD.iso     Продолжить закачку с того места,
                                          на котором она была прервана
   ...
HTTP request sent, awaiting response... 206 Partial Content
Length: 699,389,952 (667), 691,513 (66M) remaining [text/plain]
```

Благодаря функции продолжения закачки (-c) команда wget может быть особенно полезна пользователям, которые имеют низкоскоростное подключение к Интернету, а им приходится скачивать файлы больших размеров. Если вы хотя бы однажды сталкивались с разрывом соединения после многочасового процесса загрузки, то знаете, что мы имеем в виду.

ПРИМЕЧАНИЕ

Обратите внимание, что если вы не использовали параметр -c для продолжения загрузки, то файл будет сохранен под другим именем: к исходному имени добавляется .1.

Передача файлов с использованием curl

Клиент приложения для работы с URL (команда curl) предоставляет схожие с wget функциональные возможности по передаче файлов по FTP-протоколу и Интернету.

Однако команда curl также передает файлы с помощью наиболее популярных протоколов, таких как SSH (SCP и SFTP), LDAP, DICT, Telnet и File.

В отличие от wget, которая поддерживает большие, рекурсивные загрузки, команда curl предназначена для *однократных передач файлов*. Однако она поддерживает больше протоколов (как указано выше) и несколько приятных улучшенных функциональных возможностей. Для использования этой команды необходимо установить пакет curl.

Рассмотрим несколько интересных примеров **файловой передачи посредством команды curl**:

```
$ curl -O ftp://kernelorg.mirrors.tds.net/pub/linux/kernel/v1.0/patch[6-8].sign
$ curl -OO ftp://kernelorg.mirrors.tds.net/pub/linux/kernel/v2.6/ \
    ChangeLog-2.6.{1,4}
$ curl -O ftp://chris:MyPasswd@ftp.example.com/home/chris/fileA \
-Q '-DELE fileA'
$ curl -T install.log ftp://chris:MyPasswd@ftp.example.com/tmp/ \
 -Q "-RNFR install.log" -Q "-RNTO Xinstall.log
$ curl ftp://ftp.kernel.org/pub//  Отобразить содержимое /pub/
```

Первые две команды демонстрируют использование квадратных скобок для установки диапазона ([6-8]) и фигурных скобок для задания списка ({1,4}) символов или чисел, совпадающих с файлами.

Третья команда показывает, как можно ввести имя пользователя и пароль (chris:MyPasswd), скачать файл (fileA) с сервера, а затем удалить его, как только завершится загрузка (-Q '-DELE fileA').

Четвертый пример позволяет загрузить (-T) файл install.log на FTP-сервер. Затем удаленный файл переименовывается в Xinstall.log.

Последняя команда из представленных указывает вывести содержимое директории **/pub/** на **ftp.kernel.org**.

ередача файлов с помощью FTP-команд

В Ubuntu входит стандартный FTP-клиент (команда ftp), который работает, как и в большинстве систем UNIX и Windows. Мы же рекомендуем вам использовать полнофункциональную и практичную команду lftp.

FTP-клиенты позволяют открыть сессию с FTP-сервером (в отличие от wget и curl, которые просто берут отдельный файл). Затем вы управляете сервером, как и локальной файловой системой, получая и отправляя файлы посредством сетевого подключения.

Рассмотрим несколько примеров о том, как **присоединиться к FTP-серверу с помощью команды lftp**:

```
$ lftp mirrors.kernel.org             Анонимное подключение
lftp mirrors.kernel.org:~>
$ lftp francois@example.com           Подключение с проверкой подлинности
lftp example.com:~>
$ lftp -u francois example.com        Подключение с проверкой подлинности
Password: ******
```

```
lftp example.com:~>
$ lftp -u francois,Mypwd example.com     Проверка подлинности с помощью пароля
lftp example.com:~>
$ lftp                                   Запустить lftp без установки подключения
lftp :~> open mirrors.kernel.org         Установить подключение в рамках сессии lftp
lftp mirrors.kernel.org:~>
```

ВНИМАНИЕ

Не рекомендуется применять четвертый пример на практике. Пароли, вводимые в командной строке, записываются и хранятся как обычный текст в файле ~/.bash_history. Другие пользователи также могут увидеть их в результатах выполнения команды ps auwx.

После того как установлено соединение с FTP-сервером, можно использовать набор команд для FTP-сессии. Команды FTP похожи на те, что применяются в командном процессоре. Как и в оболочке bash shell, вы можете нажать клавишу Tab для автозаполнения имен. При наличии установленной сессии команда lftp поддерживает перевод нескольких процессов в фоновый режим работы (Ctrl+Z) и их возврат в состояние высокого приоритета (wait или fg). Эти функциональные возможности полезны, если вы хотите продолжить просмотр FTP-сайта, в то время как ведется загрузка файлов. Фоновые процессы выполняются параллельно. Наберите jobs, чтобы увидеть список действующих фоновых процессов. Для вывода списка команд lftp введите help.

Следующая типовая lftp-сессия демонстрирует **полезные команды для загрузки информации**:

```
$ lftp mirrors.kernel.org
lftp mirrors.kernel.org:~> pwd           Проверить текущую директорию
ftp://mirrors.kernel.org
lftp mirrors.kernel.org:~> ls            Отобразить текущую директорию
drwxr-sr-x 8 400 400 4096 Jul 02 20:19 debian/
drwxr-xr-x 7 537 537 77 May 21 21:37 fedora/
  ...
lftp mirrors.kernel.org:~> cd fedora/releases/7/Live/i386   Сменить папку
lftp mirrors.kernel.org:...> get Fedora-7-Live-i686.iso     Скачать файл
Fedora-7-Live-i686.iso at 776398 (1%) 467.2K/s eta:26m {Receiving data]
lftp mirrors.kernel.org:...> <Ctrl+Z>              Переключить загрузку
                                                   в фоновый режим
lftp mirrors.kernel.org:...> mget /gnu/ed/*        Скачать все файлы из /gnu/ed
lftp mirrors.kernel.org:...> !ls                   Локальный запуск ls
lftp mirrors.kernel.org:...> bookmark add Live     Создать закладку
lftp mirrors.kernel.org:...> quit                  Закрыть lftp
```

В начале работы сессии вы входите на сайт mirrors.kernel.org под именем анонимного пользователя. Перейдя в директорию, содержащую искомый ISO-образ, вы загружаете его с помощью команды get. Нажатие сочетания клавиш Ctrl+Z дает возможность загрузке продолжиться, в то время как вы можете заниматься другими делами. Далее команда mget (которая поддерживает подстановочные символы, такие как *) загружает все файлы из папки /gnu/ed.

Если перед любой командой поставить восклицательный знак (например, !ls), то она будет выполнена локальным командным процессором. Команда bookmark

сохраняет текущий путь (в данном случае ftp://mirrors.kernel.org/ fedora/releases/7/Live) под именем Live, чтобы в следующий раз я мог набрать lftp Live и вернуться в то же место. Команда quit завершает сессию.

Рассмотрим еще несколько **полезных команд, применяемых во время авторизированной сессии lftp, в рамках которой выполняется загрузка данных на сервер**. Предполагается, что у вас есть необходимые полномочия доступа к файлам на сервере:

```
$ lftp chris@example.com
Password: *******
lftp example.com:~> lcd /home/chris/songs       Перейти к локальной папке
lftp example.com:~> cd pub/uploads              Перейти к папке на сервере
lftp example.com:~> mkdir songs                 Создать директорию на сервере
lftp example.com:~> chmod 700 songs             Изменить права удаленной папки
lftp example.com:~> cd songs                    Перейти в новую директорию
lftp example.com:~> put song.ogg tune.ogg       Загрузить файлы на сервер
3039267 bytes transferred
lftp example.com:~> mput /var/songs/*           Загрузить соответствующие
                                                условиям файлы
lftp example.com:~> quit                        Закрыть lftp
```

Сессия lftp показывает, как можно использовать команды командной оболочки для работы с удаленными папками (подразумевается, что у вас есть необходимые права). Команды mkdir и chmod создают директорию и открывают доступ к ней только для вашей учетной записи. Команда put загружает один или несколько файлов на удаленный сервер. У команды mput есть возможность работы с подстановочными символами, чтобы одновременно загрузить множество файлов. В другие команды входят параметры mirror (чтобы скачать с сервера дерево каталогов) и mirror -R (для загрузки дерева каталогов на сервер).

Команда lftp также предоставляет **сценарий командного процессора для неинтерактивных сессий**: lftpget. Синтаксис команды lftpget похож на синтаксис wget:

```
$ lftpget ftp://mirrors.kernel.org/ubuntu/dists/feisty/Release
```

Помните, что стандартные FTP-клиенты небезопасны, так как работают с помощью чистого текста. Если вопрос безопасности стоит не на последнем месте, то альтернативным способом передачи файлов являются SSH-утилиты.

Использование инструментов SSH для передачи файлов

Инструменты SSH — одни из самых важных инструментов системного администратора, которые применяются для установки связи. В гл. 13 описаны наиболее сложные аспекты их настройки и использования. Однако в своей самой простой форме SSH-утилиты наиболее часто используются для передачи файлов.

В частности, команда scp обеспечит вас всем необходимым для передачи файла с одного компьютера на другой, в то время как она будет безопасно зашифрована на этапе авторизации и самого процесса передачи. Команду rcp заменяет ssh, став-

шая наиболее популярной утилитой для передачи файлов с одного хост-компьютера на другой.

ВНИМАНИЕ

Команда scp не сообщает о перезаписи файлов (то есть о стирании старого и записи нового), поэтому убедитесь, что целевой хост-компьютер не содержит нужных вам файлов или папок, путь которых совпадает с маршрутом указанной файловой передачи.

Копирование файлов в других узлах сети с помощью scp

Чтобы использовать команду scp для передачи файлов, на удаленной системе должен быть запущен сервис SSH (обычно это демон sshd). Рассмотрим несколько примеров **применения команды scp**:

```
$ scp myfile francois@server1:/tmp/     Скопировать myfile на server1
Password: ******
$ scp server1:/tmp/myfile .             Скопировать myfile, находящийся на другом
                                        узле, в текущую локальную папку
Password: ******
```

Используйте параметр -p для **сохранения прав доступа и временных меток** на копируемых файлах:

```
$ scp -p myfile server1:/tmp/
```

Если SSH-сервис работает не через порт 22, используемый по умолчанию, то используйте параметр -P, чтобы **указать нужный порт** в поле команды scp:

```
$ scp -P 12345 myfile server1:/tmp/     Подключиться через определенный порт
```

Для **создания рекурсивных копий** с определенного места в удаленной файловой системе используйте параметр -r:

```
$ scp -r mydir francois@server1:/tmp/   Копирует все папки mydir
                                        в удаленную папку /tmp
```

Хотя эффективность команды scp высока, когда вы точно знаете местонахождение нужного файла для копирования, иногда полезнее просматривать и передавать файлы интерактивно.

Копирование файлов, находящихся на другом узле связи, в рамках сессий sftp и lftp

Команда sftp позволяет использовать интерфейс, подобный FTP-клиентам, для **поиска и копирования файлов по протоколу SSH**. Ниже приведен пример, показывающий, как начать sftp-сессию:

```
$ sftp chris@server1
chris@server1's password: *****
sftp>
```

Используйте sftp, как и обычные FTP-клиенты. Наберите ?, чтобы увидеть список команд. Вы можете переходить по удаленным (cd) и локальным папкам (lcd),

проверять их (pwd и lpwd), а также просматривать локальное и удаленное содержимое (ls и lls). В зависимости от прав доступа пользователя, под именем которого вы находитесь в системе, можно создавать и удалять папки (mkdir и rmdir), изменять права доступа (chmod) и принадлежность/группу (chown и chgrp) файлов или папок.

Вы также можете применять команду lftp (это уже обсуждалось выше) как sftp-клиент. Lftp обладает некоторыми практичными функциональными возможностями, например позволяет **завершить ввод пути** нажатием клавиши **Tab**:

```
$ lftp sftp://chris@server1
Password: ********
lftp chris@server1:~>
```

Утилиты для передачи файлов в Windows

Зачастую пользователям необходимо получить файлы с Linux-серверов, используя Windows-клиенты. Если вы работаете в Windows, то можете использовать следующие общедоступные утилиты для получения файлов с серверов Linux:

- WinSCP (**http://winscp.net**) — графический клиент scp, sftp и FTP для Windows с использованием протоколов SSH1 и SSH2;
- FileZilla (**http://filezilla.sourceforge.net**) — предоставляет графический клиент FTP- и SFTP-сервисов в Windows наряду с возможностями FTP-сервера;
- PSCP (**www.chiark.greenend.org.uk/~sgtatham/putty/**) — клиент scp, использующий командную строку, являющийся частью модуля PuTTY;
- PSFTP (**www.chiark.greenend.org.uk/~sgtatham/putty/**) — клиент sftp, использующий командную строку, являющийся частью модуля PuTTY.

Предоставление общего доступа к удаленным директориям

Средства, уже описанные в этой главе, предоставляют доступ к файлам на элементарном уровне, когда после подключения файлы передаются в рамках одного сеанса. Сервисы по предоставлению доступа и монтированию удаленных файловых систем могут быть весьма полезны, когда необходим постоянный и непрерывный доступ к удаленной папке с файлами. Такие сервисы включают в себя Сетевую файловую систему (NFS), Samba и SSHFS.

NFS

Если предположить, что на сервере уже запущен NFS-сервис (как часть установочного пакета nfs-kernel-server), то можно использовать команды exportfs и showmount, чтобы увидеть доступные и смонтированные общие папки. Монтирование общей директории выполняется с помощью специальных параметров команды mount. Если вы установите пакет программ nfs-kernel-server, Ubuntu запустит NFS-сервис.

Просмотр и экспорт общих ресурсов

Команда exportfs, запущенная на NFS-сервере, **показывает все общие папки**, которые доступны на нем:

```
$ sudo /usr/sbin/exportfs -v
/export/myshare client.example.com(ro,wdelay,root_squash,no_subtree_check)
/mnt/public            <world>(rw,wdelay,root_squash,no_subtree_check)
```

В данном примере папки с общим доступом — export/myshare и /mnt/public. Первая из них доступна только хост-компьютеру client.example.com, в то время как вторая доступна всем. Параметры для каждого общего ресурса указаны в скобках. Первая папка доступна только для чтения (ro), записи в папку откладываются, если ожидается активность, чтобы увеличить производительность (wdelay), запросы root-пользователя к клиенту рассматриваются как анонимные запросы (root_squash). К тому же проводится менее детальная проверка прав доступа к файловой системе (no_subtree_check). Во втором общем каталоге можно проводить как чтение, так и запись (rw).

Вы можете **создавать и редактировать общие NFS-директории**, внося изменения в файл /etc/exports. Для того чтобы изменения вступили в силу, в качестве root-пользователя наберите следующие команды:

```
$ sudo /etc/init.d/nfs-kernel-server reload    Перезагрузить экспортированные
                                               общие каталоги
$ sudo exportfs -r                 То же самое
$ sudo exportfs -rv                Полная перезагрузка экспортированных общих
                                   каталогов
exporting client.example.com:/export/myshare
exporting *:/mnt/public
```

Запустив команду showmount на сервере Linux, можно **увидеть список доступных директорий на локальной системе**. Например:

```
$ sudo /usr/sbin/showmount -e
Export list for server.example.com
/export/myshare       client.example.com
/mnt/public                  *
```

Если же вы запустите команду showmount на клиентской Linux-системе, то можете **увидеть список доступных директорий на выбранном компьютере**. Например:

```
$ sudo /usr/sbin/showmount -e server.example.com
/export/myshare       client.example.com
/mnt/public           *
```

Монтирование общих ресурсов

Используйте команду mount, чтобы смонтировать удаленный общий ресурс NFS на локальном компьютере. Приведем соответствующий пример:

```
$ sudo mkdir /mnt/server-share
$ sudo mount server.example.com:/export/myshare /mnt/server-share
```

В рамках данного примера мы имеем дело с NFS-сервером (server.example.com) и общей папкой с этого сервера (/export/myshare). В конце команды указывается локальная точка монтирования (/mnt/server-share), которая должна существовать до самого процесса монтирования общей папки.

Для **указания специальных NFS-свойств** добавьте параметр -o после команды mount:

```
$ sudo mount -o rw,hard,intr server.example.com:/export/myshare /mnt/server-share
```

Параметр rw позволяет монтировать удаленную папку с правами доступа чтение/запись при условии, что такие права доступны. Если указан параметр hard, то любому, кто будет использовать общий ресурс, будет выдаваться сообщение Server not responding (здесь: Сервер не отвечает), как только будет достигнут лимит времени на операцию чтения или записи. В таких случаях полезно указать параметр intr, который позволяет прервать запрос на разрыв соединения с удаленным сервером (нажмите сочетание клавиш Ctrl+C).

Протокол NFS версии 3 (nfs3) применяется по умолчанию для подключения к общему ресурсу. Для использования NFS версии 4, созданной для работы через Интернет и брандмауэры, укажите этот протокол в командной строке как тип файловой системы следующим образом:

```
$ sudo mount -t nfs4 server.example.com:/ /mnt/server-share
```

ПРИМЕЧАНИЕ

В зависимости от используемой версии Ubuntu протокол NFS v4 может работать со сбоями. Работа с более ранними версиями NFS через SSH может оказаться намного надежнее и/или безопаснее. Можете найти дополнительную информацию об этом через Интернет, набрав в строке поиска nfs ssh.

В частности, обратите внимание на сайт www.howtoforge.com/nfs_ssh_tunneling и загляните по адресу http://tldp.org/HOWTO/NFS-HOWTO/security.html для получения дополнительной информации по безопасности работы NFS.

Samba

Samba — это свободно распространяемая версия протокола общего доступа к файлам и принтерам в Windows. Этот протокол ранее был известен под названием Server Message Block (SMB). Теперь его называют Common Internet File System (CIFS). Samba реализована как в Linux, так и в других операционных системах. Установите пакеты программ samba и samba-doc, чтобы иметь возможность пользоваться рассматриваемой версией.

Samba SWAT — это графическая веб-утилита Windows для монтирования, предоставления общего доступа и опроса общих SMB-директорий. Чтобы **иметь возможность пользоваться SWAT в Linux**, установите пакет swat. Далее прочтите инструкции по запуску SWAT на сайте https://help.ubuntu.com/community/Swat.

Команды по работе с общими ресурсами Samba могут применяться для опроса SMB-серверов, а также для монтирования и предоставления общего доступа к папкам.

Подключение и просмотр общих ресурсов

Если вам нужно **просканировать сеть на наличие SMB-хостов**, введите следующее:

```
$ findsmb
                                       *=DMB
                                       +=LMB

IP ADDR          NETBIOS NAME          WORKGROUP/OS/VERSION
---------------------------------------------------------------------
192.168.1.1      SERVER1               +[MYWORKGROUP] [Unix] [Samba 3.0.25a-3.fc7]
```

Для **просмотра текстовой информации о составе сети** (общие папки и принтеры) используйте команду smbtree:

```
$ sudo smbtree
Password: ******
MYGROUP
  \\THOMPSON                  Samba Server Version 3.0.25a-3.fc7
   \\THOMPSON\hp2100          HP LaserJet 2100M Printer
   \\THOMPSON\IPC$            IPC Service (Samba Server Version 3.0.25a-3.fc7)
  \\EINSTEIN                  Samba Server
   \\EINSTEIN\hp5550          HP DeskJet 5550 Printer
   \\EINSTEIN\IPC$            IPC Service (Samba Server)
```

Для **создания нового пользователя Samba, идентичного существующему пользователю Linux**, используйте команду smbpasswd:

```
$ sudo smbpasswd -a francois
New SMB password: ******
Retype new SMB password: ******
Added user francois
```

ВНИМАНИЕ

Вам необходимо задать собственный Samba-пароль, чтобы выполнять команды, требующие введения пароля.

Чтобы **получить список служб, доступных анонимному пользователю**, введите следующие команды:

```
$ smbclient -L server
Password: ******
Anynymous login successful
Domain=[MYGROUP] OS=[Unix] Server=Samba 3.0.25a-3.fc7
tree connect failed: NT_STSTUS_LOGON_FAILURE
```

После этого вы можете увидеть результат использования команды smbclient от лица пользователя **francois**:

```
$ smbclient -L server -U francois
Password: ******
Domain=[MYGROUP] OS=[Unix] Server=[Samba 3.0.25a-3.fc7]
   Sharename    Type   Comment
```

```
---------    ----  -------
IPC$              IPC        IPC Service (Samba Server Version 3.0.25a-3.fc7)
hp5550      Printer HP DeskJet 5550 Printer
      Server            Comment
      ---------         -------
      THOMPSON          Samba Server Version 3.0.25a-3.fc7
      Workgroup         Master
      ---------         -------
      MYGROUP           THOMPSON
```

Для **подключения к Samba как к FTP** наберите следующую команду:

```
$ smbclient //192.168.1.1/myshare -U francois
Password:
Domain=[MYWORKGROUP] OS=[Unix] Server=[Samba 3.0.25a-3.fc7]
smb: \>
```

Как и в большинстве FTP-клиентов, чтобы увидеть список возможных команд, наберите ?. Более того, вы можете использовать команды, подобные командам командного процессора. Например, cd, ls, get, put и quit для **работы с хост-компьютером SMB**.

Монтирование общих ресурсов

Вы можете **смонтировать удаленные каталоги Samba на своем локальном компьютере** подобно тому, как это делается с локальной файловой системой или удаленной файловой системой NFS. Для монтирования общего ресурса необходимо ввести следующее:

```
$ sudo mount -t smbfs -o username=francois,password=MySecret \
 //192.168.1.1/myshare /mnt/mymount/
```

ПРИМЕЧАНИЕ

Файловая система Samba (smbfs) больше не существует в некоторых дистрибутивах Linux; однако Ubuntu поддерживает именно ее. В других дистрибутивах во время монтирования удаленного ресурса Samba предпочтительно указать CIFS (-t cifs) в качестве файловой системы.

Вы можете **отобразить текущие подключения и блокировки файлов** на сервере, используя команду smbstatus. Если кто-либо смонтировал ваши общие папки или в данный момент подключен к вашему серверу с помощью команды smbclient, команда smbstatus сообщит об этом:

```
$ sudo smbstatus
Samba version 3.0.25a-3.fc7
PID     Username     Group        Machine
-------------------------------------------
5466 francois     francois     10.0.0.55    (10.0.0.55)
Service       pid      machine       Connected at
--------------------------------------------
myshare   5644        10.0.0.55    Tue Jul 3 15:08:29 2007

No locked files
```

Поиск хост-систем

В Samba для идентификации хост-систем используются имена NetBIOS. Вы можете **определить IP-адрес компьютера** с помощью команды nmblookup, применяемой для пересылки сигналов определенному имени NetBIOS в рамках локальной подсети:

```
$ nmblookup thompson
querying thompson on 192.168.1.255
192.168.1.1 server1<00>
```

Чтобы **найти IP-адрес сервера в определенной подсети**, используйте параметр -U:

```
$ nmblookup -U 192.168.1.255 server1
querying server1 on 192.168.1.255
server1<00>
```

Проверка конфигурации

Если вы не можете пользоваться общим ресурсом Samba или появились проблемы со связью с Samba-сервером, можете проверить конфигурацию Samba на сервере. Команда testparm применяется для **просмотра главного файла конфигурации Samba** (smb.conf):

```
$ testparm
Load smb config files from /etc/samba/smb.conf
Processing section "[homes]" Processing section "[printers]" Processing section
"[myshare]" Loaded services file OK.
Server role: ROLE_STANDALONE
Press Enter to see a dump of your service definitions
```

После нажатия клавиши **Enter** (как предписывает программа) вы увидите настройки из файла smb.conf. Ниже показано, как запись об общем каталоге **myshare** (он использовался в приведенном выше примере) может выглядеть в файле smb.conf:

```
[myshare]
        path = /home/francois
        username = francois
        valid users = francois
        hosts allow = einstein
        available = yes
```

Такая запись открывает пользователю **francois** доступ к папке /home/francois (представленной общим именем **myshare**) на хост-компьютере под названием **einstein**. На данный момент общий ресурс отмечен как доступный.

Предыдущий пример применения команды testparm показывает записи, помещаемые в файл smb.conf. Однако здесь **не рассматриваются значения, используемые по умолчанию**, которые не установлены. Их можно просмотреть с помощью параметра -v. Используйте его вместе с командой less, чтобы иметь возможность постраничного просмотра информации:

```
$ testparm -v | less
```

Если вы хотите **провести проверку файла конфигурации перед запуском**, то у команды testparm есть возможность использовать файл, отличный от /etc/samba/smb.conf:

```
$ testparm /etc/samba/test-smb.conf
```

SSHFS

Монтирование удаленных файловых систем — еще один волшебный прием, который выполняется с помощью протокола SSH. Вы можете смонтировать любую директорию SSH-сервера, к которой есть доступ из вашей локальной системы Linux под определенной учетной записью пользователя, с помощью файловой системы SSH (sshfs). Она проводит шифрование операции монтирования, а также любой передаваемой информации. Еще одним потрясающим аспектом системы sshfs является то, что она не требует никакой установочной части на стороне сервера (в отличие от работы SSH-сервиса).

Вашему вниманию предоставляется быстрая процедура **монтирования удаленной папки с документами на локальный компьютер**. Для проведения этой процедуры необходимо, чтобы удаленный сервер был доступен, на нем была запущена SSH и целевая директория была открыта для вашей учетной записи пользователя. В этом примере мы монтируем директорию /var/docs с хост-компьютера с IP-адресом 10.0.0.50 в точку монтирования /mnt/docs на локальной системе:

```
$ sudo apt-get install sshfs      Установить программное обеспечение sshfs
$ sudo mkdir /mnt/docs            Создать точку монтирования
$ sudo sshfs chris@10.0.0.50:/var/docs /mnt/docs     Смонтировать удаленную папку
```

Закончив пользоваться удаленной папкой, вы можете **демонтировать** ее с помощью команды fusermount (часть установочного пакета fuse-utils):

```
$ sudo fusermount -u /var/docs    Демонтировать удаленную папку
```

Общение с друзьями через IRC

Несмотря на всенародную любовь к клиентам передачи мгновенных сообщений, интернет-чаты (Internet Relay Chat, IRC) все еще остаются популярными. На сайте freenode.net есть множество комнат для общения, посвященных поддержке крупных проектов по разработке программного обеспечения с открытым исходным кодом. Более того, множество людей находятся там на протяжении всего дня и наблюдают за дискуссиями, касающимися их любимых Linux-проектов. Этот феномен известен под названием lurking и означает пассивный просмотр сетевых новостей и конференций.

Утилита xchat — хороший графический, многофункциональный IRC-клиент. Вы можете установить пакет программ xchat или модули GNOME из установочного пакета xchat-gnome. Если вы установили последний, то в Ubuntu выберите Applications ▸ Internet ▸ XChat-GNOME IRC Chat (Приложения ▸ Интернет ▸ Чат GNOME IRC). Однако элитным способом общения в IRC является использование текстового клиента в окне терминала на постоянно работающей машине, например

на старом сервере. Другим похожим вариантом является работа с IRC прокси-клиента, также известного как баунсер, например программы dircproxy (часть установочного пакета dircproxy).

Исходным IRC-клиентом был ircII. Он поддерживал сценарии наподобие макросов в комплектах приложений для продуктивной работы, которые автоматизировали некоторые команды и повышали практичность программы. Наиболее популярным был PhoEniX от Vassago, затем появился BitchX, который сначала был сценарием ircII, а затем превратился в отдельный клиент. Большинство людей на сегодняшний день используют irssi. Для того чтобы **установить и запустить irssi** из Ubuntu, наберите следующее:

```
$ sudo apt-get install irssi
$ irssi -n JayJoe199x
```

В этом примере именем пользователя является JayJoe199x (вы должны выбрать свое имя). Вы увидите синюю строку состояния, которая означает, что вы находитесь в окне 1 (окно состояния). После первого запуска irssi программа выведет справочное сообщение, ссылающееся на документацию. Всем IRC-командам предшествует символ / Например, чтобы **подключиться к серверу freenode**, наберите следующее:

```
/connect chat.freenode.net
```

Если вы не указали имя пользователя в командной строке, то войдете в чат сайта freenode.net под именем текущего пользователя. В IRC комнаты общения называются *каналами* и перед именем каждого стоит знак решетки (#).

Теперь попытайтесь **присоединиться к IRC-каналу #centos**:

```
/join #centos
```

Ваш экран должен быть похож на тот, что вы видите на рис. 12.1.

```
00:20 -!- Irssi: #centos: Total of 226 nicks (5 ops, 0 halfops, 0 voices, 221 normal)
00:20 -!- Channel #centos created Sat Nov 25 22:42:39 2006
00:22  JayJoe199x  hello peeps
```

Рис. 12.1. Клиент irssi подключен к каналу #centos на Freenode

ПРИМЕЧАНИЕ

Являясь пользователем Ubuntu, вы, возможно, захотите подключиться к каналу #ubuntu.

Как указано в строке состояния, теперь вы находитесь в канале в окне 2. Переключайтесь между окнами с помощью сочетаний клавиш Alt+1, Alt+2 и т. д. (или Ctrl+N и Ctrl+P). Обратите внимание, что такие сочетания клавиш, как Alt+1 и Alt+2, не работают в окне терминала gnome, так как он поглощает их. Для **получения помощи** наберите /help «команда», где под командой подразумевается любая команда,

о которой вы хотите узнать подробнее. Текст будет выведен в окне состояния (но не обязательно в текущем окне).

Для **отправки сообщения** в IRC-чат просто наберите его и нажмите клавишу Enter, после чего сообщение увидят все пользователи канала. Для получения дополнительной информации об использовании клиента irssi обращайтесь к документации по адресу www.irssi.org/documentation.

Работа с почтовыми программами на основе текстовых приложений

В наши дни большинство почтовых агентов пользователя (MUA) работают на основе графического интерфейса. Таким образом, если вы начали пользоваться электронной почтой в последнем десятилетии (или около того), то, подумав о почтовой программе, вы в первую очередь представите себе Evolution, Kmail, Thunderbird или (в системе Windows) Outlook. Однако в первых версиях UNIX и Linux почта обрабатывалась на основе текстовых приложений.

Если у вас возникнет необходимость проверить электронную почту на удаленном сервере или в рамках другой текстовой среды, то предлагаем вашему вниманию достойные текстовые почтовые программы. К слову, многие компьютерные фанаты до сих пор используют только текстовые почтовые программы, расхваливая их эффективность и посмеиваясь над сообщениями на основе HTML.

При описании почтовых программ мы подразумевали, что ваши сообщения хранятся в локальной системе в стандартном формате MBOX. Это означает что либо вы подключены к почтовому серверу, либо уже локально провели загрузку сообщений (например, с помощью POP3 или подобных средств).

ПРИМЕЧАНИЕ

Текстовые почтовые программы можно использовать для чтения сообщений, уже сохраненных другими почтовыми программами. Например, можно открыть файл входящего сообщения программы Evolution с помощью следующей команды: mail -f $HOME/.evolution/mail/loc/Inbox.

Команда mail

Для быстрой проверки почты в почтовом ящике суперпользователя на удаленном сервере чаще всего применяют команду `mail` (`/bin/mail`), которая является частью установочного пакета mailx. Хотя существует возможность ее интерактивного использования, наиболее часто ее применяют для **отправки электронных сообщений на основе сценариев**. Рассмотрим несколько примеров:

```
$ mail -s 'My Linux version' chris@example.com < /etc/lsb-release
$ ps auwx | mail -s 'My Process List' chris@example.com
```

Эти примеры демонстрируют быстрый способ отправки текста без необходимости открывать графическое приложение. Первый пример позволяет отправить содержимое файла `/etc/lsb-release` пользователю по адресу chris@example.com. Темой (`-s`) является `'My Linux Version'`. Во втором примере показывается, как тому

же пользователю отправляется список текущих запущенных приложений (ps auwx) с темой 'My Process List'.

ПРИМЕЧАНИЕ

Перед тем как вы сможете использовать почтовую программу командной строки, нужно настроить пакет. Существует множество аспектов настройки почтовых серверов, зависящих от вашего интернет-провайдера. Процесс конфигурации начинается как часть установки благодаря следующей команде:

```
$ sudo apt-get install mailx
```

При интерактивном использовании команды mail по умолчанию открывается почтовый ящик, указанный в текущем значении $MAIL интерпретатора команд. Например:

```
$ echo $MAIL
/var/spool/mail/chris
```

ПРИМЕЧАНИЕ

Возможно, понадобится задать значение этой переменной среды. Оно должно выглядеть так: /var/spool/mail/«имя пользователя» (в текущем примере именем пользователя является chris). В Ubuntu значение переменной MAIL, как и сама команда, по умолчанию не установлено.

Чтобы прочитать почту пользователя **root**, наберите следующую команду:

```
$ sudo mail
Mail version 8.1 6/6/93. Type ? for help.
"/var/spool/mail/root": 25 messages 25 new
>U 1 logwatch@ab.1 Fri Jun 15 20:03 44/1667 "Logwatch for ab (Linux)"
U 2 logwatch@ab.1 Sat Jun 16 04:32 87/2526 "Logwatch for ab (Linux)"
3 logwatch@ab.1 Sun Jun 17 04:32 92/2693 "Logwatch for ab (Linux)"
N 4 logwatch@ab.1 Fri Jun 22 09:28 44/1667 "Logwatch for ab (Linux)"
N 5 MAILER-DAEMON@ab Fri Jun 22 09:28 93/3348 "Warning: could not send"
&
```

Перед текущим сообщением стоит знак >. Новые сообщения имеют обозначение N в начале, непрочитанные (но не новые) сообщения — U. Если же никакого символа нет, то сообщение было прочитано. Знак & внизу означает, что программа готова принимать команды.

На этом этапе вы находитесь в командном режиме. Можно использовать простые команды для **навигации и работы с основными почтовыми функциями**. Наберите ?, чтобы увидеть список команд, или введите номер сообщения, которое хотели бы прочесть. Наберите v3, чтобы увидеть третье сообщение в редакторе vi. Если хотите увидеть список заголовков сообщений начиная с 18-го, введите h18. Для ответа на сообщение под номером 7 наберите r7 (введите текст сообщения, затем поставьте точку, чтобы отправить его). Наберите d4, чтобы удалить четвертое сообщение (или d4-7, если хотите удалить сообщения, начиная с четвертого и заканчивая седьмым). Для того чтобы выйти в интерпретатор команд, введите !bash (затем exit, чтобы вернуться в приложение mail).

Перед тем как выйти из приложения, обратите внимание, что, как только вы выходите из программы, все просмотренные сообщения копируются из почтового

ящика в файл &HOME/mbox. Это происходит в том случае, если вы их не сохранили (посредством команды pre*). Чтобы все сообщения остались в почтовом ящике, выйдите из приложения, нажав клавишу X. Для сохранения изменений почтового ящика при выходе из программы нажмите клавишу Q.

Вы можете открыть любой файл формата MBOX во время пользования приложения mail. Например, если вы находитесь в системе под именем пользователя, отличным от chris, но хотите просмотреть почтовый ящик этого пользователя, введите следующее:

```
$ sudo mail -f /var/spool/mail/chris
```

Команда mutt

Если вы хотите постоянно использовать почтовый клиент из командной строки, то мы рекомендуем работать с командой mutt вместо mail. У команды mail есть множество ограничений, например в ней не реализована поддержка отправки приложений без их предварительного кодирования (в частности, с помощью команды uuencode); в то время как mutt обладает многими современными возможностями работы с электронной почты. Команда mutt является частью пакета программ mutt, который необходимо установить, чтобы иметь возможность пользоваться данной программой. Вы можете настроить работу команды в файле /etc/Muttrc. Кроме того, нужно будет настроить команду sendmail, чтобы отправлять электронную почту.

Подобно mail, mutt обладает возможностью отправки сообщений с помощью сценариев. Кроме того, она может **работать с приложениями к почте**. Например:

```
$ mutt -s "My Linux Version" -a /etc/lsb-release \
 chris@example.com < email-body.txt
$ mutt -s "My Linux Version" -a /etc/lsb-release \
 chris@example.com < /dev/null
```

В первом примере файл email-body.txt служит телом письма, а /etc/lsb-release — приложением. Во втором примере отправляется пустое письмо (< /dev/null), однако отсылается то же приложение, что и в первом примере.

Вы можете **начать сессию mutt** (если почтовым ящиком по умолчанию является $MAIL), просто набрав команду:

```
$ mutt
/home/chris/Mail does not exist. Create it? ([yes]/no)  y
q:Quit  d:Del  u:Undel  s:Save  m:Mail  r:Reply  g:Group  ?:Help
1 OJun 16 logwatch@ab       (  69) Logwatch for ab (Linux)
2 OJun 18 logwatch@ab       ( 171) Logwatch for ab (Linux)
3 OJun 18 Mail Delivery S   ( 219) Warning: could not send message
4 OJun 19 logwatch@ab       (  33) Logwatch for ab (Linux)
--Mutt: /var/spool/mail/root [Msgs:22 New:2 Old:20 63K]--(date/date)--(all)--
```

С приложением mutt легче работать, чем с mail, так как это экранно-ориентированная программа. Как и в случае с mail, **используйте клавиши, чтобы перемещаться в mutt**. Как всегда, нажмите ?, чтобы получить помощь. Советы по работе с электронной почтой появляются на верхней панели. Используйте клавиши для управления курсором ↑ и ↓ для выделения сообщений, которые хотите прочесть.

Клавиши Page Up и Page Down применяются для прокрутки страниц каждого сообщения. Нажмите клавишу I для возврата к заголовкам сообщений.

Проводите поиск текста с помощью клавиши / (искать далее) и Esc+/ (поиск в обратном направлении). Нажмите N для нового поиска. Чтобы перейти к следующему новому или непрочитанному сообщению, нажмите клавишу Tab. Перейти к предыдущему сообщению вам позволит нажатие сочетания клавиш Esc+Tab. Клавиша S используется для сохранения текущего сообщения в файл. Нажмите D для удаления сообщения, а U — для отмены удаления.

Для отправки нового электронного сообщения используйте клавишу M. После заполнения адреса получателя и темы пустое окно открывается в редакторе JOE (или другом редакторе, определенном по умолчанию в переменной `$EDITOR`). После выхода из режима создания письма нажмите клавишу A, если хотите присоединить файл (добавить приложение к письму). Нажмите клавишу ?, чтобы увидеть другие способы работы с сообщениями, заголовками и приложениями. Нажав Y, вы отправите сообщение, а Q — прервете отправку.

Закончив работу, нажмите клавишу X для выхода без сохранения изменений ящика. Если хотите выйти и сохранить все сделанные изменения (прочитанные, удаленные сообщения и пр.), нажмите клавишу Q.

Резюме

Команды сетевого доступа предоставляют быстрый и эффективный способ получения необходимой информации через сеть. Браузер elinks является популярной экранно-ориентированной программой для просмотра и получения информации в Интернете или открытия любого HTML-файла. Существуют десятки команд для загрузки файлов через протоколы FTP, SSH, HTTP и др., например `wget`, `curl`, `lftp` и `scp`.

В этой главе рассматриваются командные утилиты NFS, Samba и SSHFS, которые полезны для получения постоянного доступа к удаленным директориям и файлам. Воспользоваться популярным чатом IRC можно с помощью команды `irssi`. Работать с электронной почтой вы можете, выбрав понравившийся текстовый клиент — mail или mutt.

13 Удаленное администрирование

Большинство профессиональных системных администраторов Linux не используют графическую систему на своих интернет-серверах. Таким образом, если вы хотите получить доступ к другим компьютерам для удаленного администрирования, вам придется какое-то время работать в командной строке. К счастью, существует множество функциональных Linux-команд, которые помогут в этом.

Утилиты, связанные с безопасным командным процессором (SSH), не только позволяют получать удаленный доступ и передавать файлы, но и предоставляют функцию шифрования данных, чтобы ваша работа по удаленному администрированию была безопасной. С помощью таких утилит, как Virtual Network Computing (VNC), вы сможете запустить Рабочий стол удаленного сервера на вашем локальном компьютере. Эти и другие функции по удаленному администрированию описаны в данной главе.

Регистрация в удаленной системе и туннелирование с помощью SSH

Старшая сестра Linux — система UNIX — создавалась в университетских сетях. В то время пользователями таких изолированных друг от друга сетей были профессора и студенты, поэтому необходимости в безопасности таких сетей не было.

Приложения и протоколы, созданные в то время (1970–1980-е годы), отражают недостаточное внимание, уделяемое шифрованию и процессу регистрации в удаленной системе. SMTP является отличным примером данного явления. То же можно сказать и об утилитах для удаленной работы первого поколения: telnet, ftp (протокол FTP), rsh (удаленный интерпретатор команд), rcp (удаленное копирование), rexec (удаленное выполнение) и rlogin (удаленный вход в систему). Эти утилиты посылали данные о пользователе и трафик посредством обычного текста. По этой причине их опасно использовать в общественно доступных сетях, таких как сегодняшний Интернет. По большей части всеми этими средствами уже не пользуются — их заменили команды SSH (команды ssh, scp, sftp и родственные сервисы).

Хотя для устаревших команд по работе с удаленным доступом все еще находится применение (см. врезку «Использование устаревших средств коммуникации»), большая часть этой главы посвящена SSH-командам, с помощью которых можно удовлетворить все ваши потребности в области удаленной связи.

Использование устаревших средств коммуникации

Несмотря на тот факт, что SSH предоставляет более совершенные средства удаленной коммуникации, устаревшие команды (их иногда называют r-команды) до сих пор включаются во все крупные дистрибутивы Linux. Некоторые из этих инструментов будут работать быстрее, чем соответствующие им команды SSH, так как им не надо проводить шифрование данных. Некоторые администраторы UNIX старой закалки могут иногда пользоваться ими в личных сетях или все еще включать в свои сценарии. Хотя по большей части вы будете просто оставлять в стороне устаревшие команды удаленного доступа, в некоторых случаях telnet может быть полезной.

Команда telnet все еще применяется для связи с некоторыми сетевыми устройствами (маршрутизаторы, переключатели, UPS и пр.), у которых нет мощности для запуска демона ssh. Хотя она несет в себе некоторый риск, связанный с безопасностью, некоторые производители все еще включают поддержку telnet в свои устройства.

Отличным способом применения команды telnet является устранение неполадок связанных с интернет-протоколами POP3, SMTP, HTTP и др. На самом деле данные текстовые протоколы — это автоматические сессии telnet, в течение которых клиент (например, браузер или почтовая программа пользователя) обменивается текстом с сервером. Единственным отличием является используемый TCP-порт. Вот пример использования telnet через HTTP-порт (80) веб-сервера:

```
$ telnet www.example.com 80
Trying 208.77.188.166.
Connected to www.example.com.
Escape character is '^]'
GET / HTTP/1.0
        Введите второй символ возврата каретки
HTTP/1.1 200 OK
```

Подобно этому, вы можете настроить команду telnet на работу с портом 25 почтового сервера (SMTP) и 110 (POP3) и использовать необходимые команды для устранения неполадок с электронной почтой. Более подробно применение telnet для устранения неполадок с сетевыми протоколами описывается в книге Linux Troubleshooting Bible (Wiley Publishing, 2004).

Если вы хотите выйти из telnet-сессии, задействуйте ESC-последовательность (Ctrl+] по умолчанию). Это прекратит посылку информации с вашей клавиатуры на удаленный компьютер и выведет окно командной строки telnet, где вы можете напечатать quit для выхода или ?, чтобы увидеть все параметры.

Настройка SSH

На сегодняшний день настоящим многофункциональным инструментом сетевых администраторов является SSH. Команды и сервисы SSH заменяют все старые средства для удаленного доступа и добавляют отличное шифрование данных, открытые ключи и другие функции. Наиболее распространенным воплощением SSH в мире Linux является OpenSSH (www.openssh.com) — программа, обслуживаемая проектом OpenBSD. В OpenSSH входят клиентская и серверная части.

Для установки сервера OpenSSH выполните следующую команду:

```
$ sudo apt-get install openssh-server
```

Рассмотрим некоторые новые особенности SSH.

- В среде Windows можно использовать утилиты SSH из Linux с помощью Cygwin (www.cygwin.com). Если вы уже используете Cygwin (эмулятор среды Linux для Windows), то мы рекомендуем PuTTY (www.chiark.greenend.org/uk/sgatatham/putty). PuTTY — это мощный Telnet/SSH-клиент с открытым кодом доступа.
- Используйте SHH версии 2, где это возможно, так как она наиболее хорошо защищена. Некоторые сетевые устройства, поддерживающие SHH, могут работать только с более ранними, менее безопасными версиями. OpenSSH поддерживает все версии. Некоторые предыдущие версии Ubuntu принимали подключения SSH 1 и SSH 2. Однако новые выпуски работают с версией 2 по умолчанию.
- В Ubuntu выполните команду `/etc/init.d/ssh start`, чтобы **запустить сервис SSH** (демон sshd). Для настройки сервиса отредактируйте файл `/etc/ssh/sshd_config`.
- Чтобы **настроить клиент ssh**, отредактируйте файл `/etc/ssh/ssh_config`.

Если вы предпочитаете использовать графические утилиты для администрирования удаленной Linux-системы, то можете активировать X11-туннелирование (его также называют X11 Port Forwarding). С включенным X11-туннелированием (как на сервере, так и на клиенте) вы можете запустить приложение X на сервере, и оно будет отображаться на клиенте. Вся передаваемая через это подключение информация зашифрована.

В Ubuntu переадресация портов X11 включена (`X11Forwarding yes`) на сервере посредствам демона sshd. Вам все же необходимо включать ее на стороне клиента. Чтобы **включить переадресацию X11 на клиенте в рамках одной сессии**, подключитесь с помощью следующей команды:

```
$ ssh -X francois@myserver
```

Чтобы **включить переадресацию X11 на постоянной основе для всех пользователей**, добавьте строку `ForwardX11 yes` в файл `/etc/ssh/ssh_config`. Чтобы переадресация X11 постоянно была активна для определенного пользователя, добавьте строку в файл этого пользователя `~.ssh/config`. Как только эти установки были заданы, параметр `-X` больше не нужен для запуска X11-туннелирования. Выполните команду `ssh`, как обычно, для подключения к удаленной системе. Для проверки работы туннелирования после установки соединения с удаленной машиной с по-

мощью ssh запустите команду xclock, и данное приложение запустится на **Рабочем столе** вашего клиента.

SSH-туннелирование — это отличный способ безопасного использования удаленных графических утилит!

Использование команды ssh для удаленного входа в систему

Для **безопасного входа в удаленную систему** вы можете использовать один из двух возможных синтаксисов указания имени пользователя:

```
$ ssh -l francois myserver
$ ssh francois@myserver
```

Однако команды scp и sftp, рассмотренные в гл. 12, поддерживают только синтаксис user@server, поэтому мы рекомендуем привыкнуть именно к нему. Если вы не укажете имя пользователя, то ssh попытается подключиться с тем именем, под которым вы находитесь в системе. Если во время подключения вы захотите **самостоятельно прервать ssh-сессию**, наберите ESC-последовательность (~).

Доступ к SSH через другой порт

По причинам, связанным с безопасностью, удаленный хост-компьютер может указать другой порт для работы SSH-сервиса, нежели порт 22, который используется по умолчанию. При таких обстоятельствах используйте параметр -p для связи с этим сервисом:

```
$ ssh -p 12345 francois@turbosphere.com  Подключиться к SSH через порт 12345
```

Использование SSH для туннелирования (X11 Port Forwarding)

Если SSH-туннелирование настроено, как показано выше, то сервис SSH перенаправляет клиенты X Window System на ваш локальный монитор. Однако туннелирование можно использовать и с другими TCP-протоколами.

Туннелирование для клиентов X11. Следующая последовательность команд демонстрирует запуск SSH-сессии, а затем открытие нескольких X-приложений, которые должны появиться в вашей локальной рабочей области:

```
$ ssh francois@myserver                   Открыть ssh-подключение к myserver
francois@myserver's password: *******
[francois@myserver ~}$ echo $DISPLAY      Показать текущее значение экрана X
localhost:10.0                     SSH устанавливает $DISPLAY как localhost:10.0
[francois@myserver ~}$ xeyes&                   Показать мобильные «глаза экрана»
[francois@myserver ~}$ gnome-cups-manager&      Настроить удаленные принтеры
[francois@myserver ~}$ gksu services-admin&     Изменить системные сервисы
```

Туннелирование для удаленного администрирования принтеров CUPS. X11 — это не единственный протокол, который работает с переадресацией. Вы можете **задать переадресацию на любой TCP-порт** с помощью SSH. Это отличный способ

быстрой и легкой настройки безопасных туннелей. На стороне сервера не требуется никакой настройки.

Например, myserver является сервером принтеров с включенным пользовательским веб-интерфейсом сервиса CUPS (работающим через порт 631). Этот графический интерфейс доступен только с локальной машины. На текущем клиентском компьютере мы создаем туннель к этому сервису с помощью команды ssh со следующими параметрами:

```
$ ssh -L 1234:localhost:631 myserver
```

Этот пример устанавливает переадресацию порта 1234 клиентской части на порт 631 на сервере. Теперь мы можем открыть http://localhost:1234 на компьютере-клиенте. Этот запрос будет перенаправлен команде cupsd, которая ожидает сигнала через порт 631 на сервере.

Переадресация интернет-сервисов. Рассмотрим еще один пример использования SSH-туннелирования. Когда ваша **локальная машина не имеет доступа к Интернету**, но может подключиться к другому компьютеру (myserver) с активным интернет-соединением. Следующий пример позволяет посетить сайт Google.com (HTTP, TCP порт 80) через SSH-подключение к компьютеру по имени myserver, который подключен к Интернету:

```
$ ssh -L 12345:google.com:80 myserver
```

При использовании этого примера любое подключение к локальному порту 12345 переадресовывается через SSH-туннель к myserver, который, в свою очередь, открывает подключение к Google.com через порт 80. Теперь вы можете зайти на http://localhost:12345 и использовать myserver как ретранслятор сайта Google.com. Поскольку вы используете команду shh для переадресации порта, а не для получения интерпретатора команд на сервере, то можете добавить параметр -N, чтобы **предотвратить выполнение удаленных команд**:

```
$ ssh -L 12345:google.com:80 -N myserver
```

Применение SSH в качестве прокси-сервера SOCKS

Предыдущий пример демонстрирует, что вы можете переадресовать порт от клиента к компьютеру, отличному от сервера. В реальности лучшим способом **вывести трафик браузера из вашей локальной сети через кодируемый туннель** является использование встроенной в SSH функции прокси-сервера SOCKS. Например:

```
$ ssh -D 12345 myserver
```

Динамический параметр (-D) ssh позволяет войти в myserver (как обычно). Пока подключение активно, все запросы, отправленные порту 12345, переадресовываются на myserver. Далее установите прокси-сервер SOCKS v5 в браузере как localhost:12345, и вы будете готовы к его использованию. Не вводите ничего в поля HTTP и других протоколов. Они все работают через SOCKS. На рис. 13.1 показано окно настройки подключений Firefox.

Рис. 13.1. Используйте окно настройки подключений FIrefox для определения параметров прокси-сервера

Для проверки настроек отключите сессию ssh и зайдите на любой сайт. Браузер должен выдать сообщение об ошибке прокси-сервера.

Выбрав команду Connection ▸ SSH ▸ Tunnels (Подключение ▸ SSH ▸ Туннели) в Putty, вы можете реализовать такую же переадресацию и в среде Windows.

SSH-аутентификация с использованием открытого ключа

До сих пор мы использовали команду ssh с аутентификацией по умолчанию. Команда также поддерживает аутентификацию с использованием открытого ключа. Это имеет несколько преимуществ.

- **Автоматический вход в систему для сценариев и процессов cron.** Установив пустую фразу-пароль, вы можете использовать ssh в сценариях для автоматического входа в систему. Хотя это и удобно, но небезопасно, так как любой, кто получит доступ к вашему файлу с ключом, может подключиться к любой машине, к которой вы имеете доступ. Настройка автоматического входа в систему также может осуществляться с помощью фразы-пароля и агента по работе с ключами. Как показано ниже, это компромисс между удобством и безопасностью.
- **Двухфакторная аутентификация.** При использовании ключа с фразой-паролем для интерактивного входа в систему аутентификация проводится по двум факторам (ключ и фраза-пароль) вместо одного.

Вход в систему с использованием открытых ключей. Рассмотрим **процесс установки связи** между двумя Linux-системами на основе ключа. В следующих

примерах мы используем пустые фразы-пароли, не применяя имя пользователя и пароль. Если вы желаете защитить ключ с помощью пароля, то просто введите его во время первого шага (создание пары ключей).

Запустите следующую команду ssh-keygen на компьютере-клиенте для **создания пары ключей**, когда находитесь в системе под именем пользователя, которому необходимо установить связь:

```
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/chris/.ssh/id_rsa): <Enter>
Enter passphrase (empty for no passphrase): <Enter>
Enter same passphrase again: <Enter>
Your identification has been saved in /home/chris/.ssh/id_rsa. Your public key has
been saved in /home/chris/.ssh/id_rsa.pub. The key fingerprint is:
ac:db:a4:8e:3f:2a:90:4f:05:9f:b4:44:74:0e:d3:db chris@host.domain.com
```

Обратите внимание, что при каждом приглашении к действию вы нажимали Enter для создания файла ключа, используемого по умолчанию и для ввода (подтверждения) пароля. Теперь у вас есть частный ключ, который должен храниться в безопасном месте, в особенности если он не был защищен паролем.

Кроме того, у вас есть открытый ключ (id_rsa.pub), который был создан предыдущей командой. Открытый ключ должен быть установлен на хост-компьютерах, к которым вы хотите подключаться. Содержимое файла ~/.ssh/id_rsa.pub нужно скопировать (безопасно) в ~/.ssh/authorized_keys2 для пользователя, который будет использовать ssh на удаленном компьютере. Файл authorized_keys2 может содержать несколько ключей, если несколько пользователей использовали ssh для подключения к этой учетной записи.

Войдите в удаленную серверную систему под именем пользователя, от имени которого хотите использовать ssh с ключом. Если у вас все еще нет папки ~/.ssh, то первым делом необходимо создать ее:

```
$ cd
$ mkdir .ssh
$ chmod 700 .ssh
```

Далее копируйте (безопасно) файл открытого ключа с клиента и поместите в файл авторизированных ключей на сервере. Это можно сделать с помощью команды scp. Предположим, что имя клиентской системы — myclient, а пользователь — chris. Введите на сервере следующее:

```
$ scp chris@myclient:/home/chris/.ssh/id_rsa.pub .    Получить клиентский
                                                      id_rsa.pub
$ cat id_rsa.pub >> ~/.ssh/authorized_keys2           Добавить к вашим ключам
$ chmod 600 ~/.ssh/authorized_keys2                   Закрыть права доступа
$ rm id_rsa.pub                                       Удалить открытый ключ после
                                                      копирования его содержимого
```

Эта процедура также может быть выполнена путем редактирования текстового файла ~/.ssh/authorized_keys2 на сервере и копирования/вставки открытого ключа с компьютера клиента. Убедитесь, что передача происходит безопасно через ssh,

и не вставляйте никаких переносов на новую строку при записи ключа. Полный ключ должен помещаться на одной строке, даже если он выходит за пределы экрана.

Затем вы можете просто выполнять команду ssh с компьютера-клиента (применяя учетные записи пользователей, для которых проводили настройку), и сервер будет использовать ключ. Если вы установите фразу-пароль, то у вас будут ее требовать, как обычный пароль.

Сохранение частных ключей для их использования с Flash-носителя. Если вы хотите **хранить свой частный ключ** в более безопасном месте, нежели жесткий диск, то можете использовать Flash-носитель (его также называют флешкой):

```
$ mv ~/.ssh/id_rsa /media/THUMBDRIVE1/myprivatekey
```

Далее, когда вы захотите **использовать ключ**, введите следующее:

```
$ ssh -i /media/THUMBDRIVE1/myprivatekey chris@myserver
```

Использование ключей с фразами-паролями более безопасно, чем применение обычных паролей, но и более затруднительно. Для облегчения работы можно использовать команду ssh-agent, чтобы хранить разблокированные ключи на время текущей сессии. Добавив разблокированный ключ в запущенный ssh-agent, вы сможете запускать команду ssh с ключом, но у вас теперь не будут каждый раз запрашивать фразу-пароль.

Чтобы увидеть, что делает команда ssh-agent, запустите ее без параметров. После запуска появится трехстрочный bash-сценарий:

```
$ ssh-agent
SSH_AUTH_SOCK=/tmp/ssh-SkEQZ18329/agent.18329; export SSH_AUTH_SOCK;
SSH_AGENT_PID=18330; export SSH_AGENT_PID;
echo Agent pid 18330;
```

Первые две строки вывода должны быть выполнены вашим интерпретатором команд. Скопируйте эти строки в командную оболочку (shell) прямо сейчас. Вы можете избежать этих действий, запустив ssh-agent и приказав интерпретатору команд bash выполнить результат работы команды. Это достигается следующим образом:

```
$ eval `ssh-agent`
Agent pid 18408
```

Теперь можно разблокировать ключи и добавлять их в запущенный агент. Допустим, вы уже создали ключ командой ssh-keygen. Теперь **добавим ключ, используемый по умолчанию**, с помощью команды shh-add:

```
$ ssh-add
Enter passphrase for /home/chris/.ssh/id_rsa: *******
Identity added: /home/chris/.ssh/id_rsa (/home/chris/.ssh/id_rsa)
```

Далее вы можете добавить ключ, хранящийся на флешке:

```
$ ssh-add /media/THUMBDRIVE1/myprivatekey
```

Для **вывода списка всех ключей, хранящихся в агенте**, используйте параметр -l:

```
$ ssh-add -l
2048 f7:b0:7a:5a:65:3c:cd:45:b5:1c:de:f8:26:ee:8d:78 /home/chris/.ssh/id_rsa
(RSA)
2048 f7:b0:7a:5a:65:3c:cd:45:b5:1c:de:f8:26:ee:8d:78
/media/THUMBDRIVE1/myprivatekey (RSA)
```

Чтобы **удалить один ключ из агента**, например находящийся на флешке, запустите команду ssh-add с параметром -d:

```
$ ssh-add -d /media/THUMBDRIVE1/myprivatekey
```

Для **удаления всех ключей, хранящихся в агенте**, используйте параметр -D:

```
$ ssh-add -D
```

Применение screen: богатый удаленный интерпретатор команд

Команда ssh открывает только одно окно. Если вы его потеряете, то можете также потерять все, что сделали на удаленном компьютере. Это может быть плачевно, если вы в тот момент работали с чем-то важным, например с двенадцатичасовой компиляцией. Если же вы хотите выполнять три команды одновременно, например vi httpd.conf, tail -f error_log и service httpd reload, необходимо открыть три отдельные ssh-сессии.

По существу, screen — это мультиплексор терминалов. Если вы системный администратор, работающий на удаленных серверах, то screen станет для вас отличной утилитой для работы с удаленным компьютером, где доступна только командная строка. Кроме того, что она позволяет создавать несколько сессий интерпретатора команд, команда дает возможность отсоединиться от него, а затем обратно подключиться к той же сессии.

Пакет программ screen установлен в Ubuntu по умолчанию.

Для **использования screen** запустите команду ssh из командной строки клиента для подключения к Linux-серверу, где установлен пакет screen. Далее просто наберите следующую команду:

```
$ screen
```

Если вы запустили команду screen из окна терминала, то сначала увидите приветствие, которое вопрошает о пицце и пиве, а затем обычную командную строку bash в окне. Для управления screen нажмите сочетание клавиш **Ctrl+A** и еще одну клавишу, например **Ctrl+A** и **?** (упоминается как Ctrl+A, ?), что приведет к отображению экрана помощи. Вот несколько команд и сочетаний клавиш, которые вы можете использовать для работы с запущенной программой screen:

```
$ screen -ls                              Список активных окон
There is a screen on:
  7089.pts-2.myserver (Attached)          Указывает на то, что окно прикреплено
1 Socket in /var/run/screen/S-francois.
$ Ctrl+A, a                               Сменить заголовок окна
```

```
Set window's title to: My Server        Набрать новый заголовок
$ Ctrl+A, c                             Создать новое окно
$ Ctrl+A, "                             Показать активные заголовки окон
Num Name           Flags
  0 My Server                           Клавиши ↑ и ↓ меняют окна
  1 bash
$ Ctrl+A, d                             Отключить окно от терминала
$ screen -ls                            Список активных окон
There is a screen on:
  7089.pts-2.myserver (Detached)        Показывает, что окно отключено
1 Socket in /var/run/screen/S-francois.
```

Только что показанная сессия screen вывела два окна (в каждом запущен интерпретатор команд shell). Вы можете создавать сколько угодно окон и называть их по своему желанию. Кроме того, вместо отключения окна от сессии screen можно просто закрыть его, выйдя из интерпретатора команд в каждом открытом окне (нажав сочетание клавиш Ctrl+D).

Когда сессия screen отключена, вы возвращаетесь в интерпретатор команд, открытый после первого входа в систему сервера. Вы можете повторно подключиться к screen-сессии так, как описано в следующем подразделе.

В табл. 13.1 показаны другие полезные сочетания клавиш для управления, доступные в screen.

Таблица 13.1. Сочетания клавиш для управления командой screen

Сочетание	Описание
Ctrl+A, ?	Показать экран помощи
Ctrl+A, C	Создать новое окно
Ctrl+A, D	Отключить окно от терминала; screen-сессия и ее окна остаются в рабочем состоянии
Ctrl+A, "	Показать список окон
Ctrl+A, '	Строка для ввода номера или имени окна, на которое необходимо переключиться
Ctrl+A, N	Просмотреть следующее окно
Ctrl+A, P	Просмотреть предыдущее окно
Ctrl+A, [	В screen отключена вертикальная прокрутка терминала. Эти клавиши включают режим прокрутки в screen. Нажмите дважды Enter для выхода
Ctrl+A, Shift+A	Переименовать текущее окно
Ctrl+A, W	Показать список имен окон в строке заголовка

Повторное подключение к сессии screen

Отключившись от сессии screen, вы можете вернуться к ней позже (даже если выйдете из системы и отключитесь от сервера). Для **повторного подключения** к сессии с одним запущенным окном наберите следующее:

```
$ screen -r
```

Если запущено несколько сессий screen, то команда screen -r не сработает. Например, следующие строки показывают, что случится, если работают две отключенные screen-сессии:

```
$ screen -r
There are several suitable screens on:
   7089.pts-2.myserver            (Detached)
   7263.pts-2.myserver            (Detached)
Type "screen [-d] -r [pid.]tty.host" to resume one of them.
```

Как и предлагает результат выполнения команды, вы можете идентифицировать сессии по их названиям (которыми по умолчанию являются комбинации ID процесса, названия tty и имени хост-компьютера). Например:

```
$ screen -r 7089.pts-2.myserver
```

Имена сессий screen

Вместо названий, используемых по умолчанию, можно **создавать более описательные имена** после запуска screen. Например:

```
$ screen -S mysession
$ screen -ls
There is a screen on:
   26523.mysession (Attached)
```

Предоставление общего доступа к сессиям

Команда screen также предоставляет общий доступ к сессиям. Эта функция прекрасно подходит для технической поддержки, так как любой подключенный к этой сессии может как печатать, так и смотреть на происходящее. Присвоение названия окну, как показано в предыдущем разделе, делает эту процедуру еще проще. В результате другой человек на другом компьютере может подключиться к серверу с помощью ssh (используя то же имя пользователя) и набрать следующее:

```
$ screen -x mysession
```

Если запущена только одна сессия screen, то по аналогии с командой screen -r вам не надо указывать имя нужного окна для подключения:

```
$ screen -x
```

Использование удаленного рабочего стола Windows

Многие системные администраторы, привыкшие использовать Linux, по возможности предпочитают администрировать свои Windows-системы именно через Linux. В ней есть такие утилиты, как rdesktop и tsclient, которые позволяют присоединиться к системе Windows, на которой запущен Windows Terminal Services.

Чтобы иметь возможность **подключиться к Рабочему столу Windows из среды Linux**, необходимо включить удаленный рабочий стол в системе Windows. Для это-

го в Windows XP, например, щелкните правой кнопкой мыши на значке **My Computer** (Мой Компьютер) и выберите пункт **Properties** (Свойства). Затем перейдите на вкладку **Remote** (Удаленное использование) и установите флажок **Allow users to connect remotely to this computer** (Разрешить удаленное подключение пользователей к этому компьютеру). Выберите пользователей, которым вы разрешаете подключаться к Windows, и нажмите кнопку **OK**.

Теперь в Linux вы можете использовать утилиты dresktop или tsclient (графический пакет на основе rdesktop) для подключения к Windows с помощью протокола удаленного рабочего стола (RDP). В Ubuntu все эти компоненты установлены.

Команда tsclient

Если вы привыкли использовать в Windows приложение **Remote Desktop Connection** (Подключение к удаленному рабочему столу) (ранее известное под названием **Terminal Services Client** (Клиент службы терминала)) для подключения одного компьютера к другому, то, возможно, оцените tsclient как отличный способ подключиться к **Рабочему столу** Windows из Linux. Запуск tsclient открывает окно **Terminal Server Client** (Клиент сервера терминала), которое симулирует интерфейс клиента удаленного доступа к **Рабочему столу** Windows.

Когда пакет программ tsclient установлен, запустите tsclient из **Рабочего стола** GNOME, выполнив команду **Applications ▸ Internet ▸ Terminal Server Client** (Приложения ▸ Интернет ▸ Клиент сервера терминала) или набрав следующую команду в своем интерпретаторе команд:

```
$ tsclient &
```

На рис. 13.2 показано окно Terminal Server Client (Клиент сервера терминала).

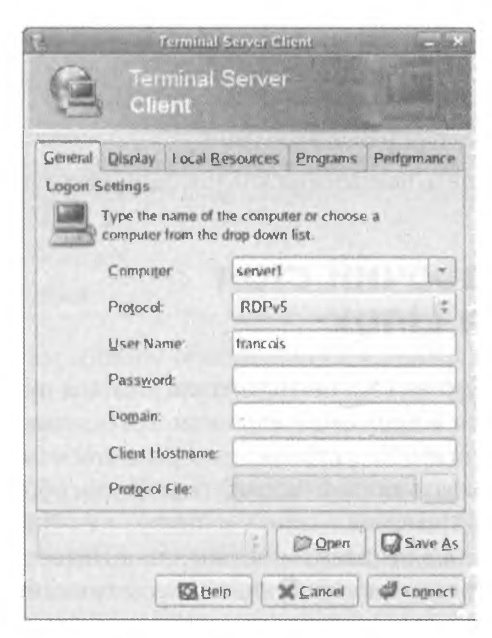


Рис. 13.2. Клиент сервера терминала (tscient) подключается к Рабочему столу Windows

Возможно, что все, что вам необходимо ввести в этом окне, — это IP-адрес системы Windows. Вероятно, у вас запросят имя пользователя и пароль, в зависимости от настроек Windows. Выбирайте различные вкладки для более детальной настройки подключения к удаленному рабочему столу Windows.

Обратите внимание на то, что tsclient также можно использовать для VNC и XDMCP.

Команда rdesktop

Если вы предпочитаете не использовать программу tclient, описанную выше, можно **войти в удаленную систему Windows** с помощью команды rdesktop. Она запрашивает вход в систему Windows, затем открывает **Рабочий стол** Windows для текущего пользователя. Вот примеры команды применения rdesktop:

```
$ rdesktop 172.16.18.66                   Вход на Рабочий стол по IP-адресу
$ rdesktop -u chris -p M6pyXX win1        Указать имя пользователя/пароль
                                          для хост-компьютера win1
$ rdesktop -f win1                        Запустить в полноэкранном режиме
$ rdesktop -0 -r sound:local win1         Прямой звук от сервера к клиенту
$ rdesktop -E win1                        Отключить шифрование клиент/сервер
```

Если вы отключите шифрование клиент/сервер, пакет с данными о входе в систему будет шифроваться, а все остальное — нет. Это может намного повысить производительность, однако любой человек, прослушивающий ваш LAN, может увидеть текст проводимых операций связи (включая все интерактивные входы в систему после первого пакета с данными о входе в систему).

Есть еще несколько параметров команды rdesktop, способных повысить производительность:

- -m — не отсылать передвижения мыши;
- -D — скрыть оформление менеджера окон;
- -K — не переопределять назначения клавиш менеджера окон.

Удаленный рабочий стол и приложения Linux

Система X Windows System (X) не должна запускаться на обычных серверах по причинам безопасности и производительности. Но благодаря характеру клиент/сервер вы можете запустить X-программу на удаленном компьютере и перенаправить ее графический вывод на свой **Рабочий стол**. Таким образом, приложение, запущенное на удаленной машине, — это X-клиент, а ваш **Рабочий стол** — X-сервер. Запуская X-приложения в ненадежных сетях или в Интернете, используйте описанную ранее SSH-переадресацию. В надежных сетях делайте это без SSH, как описано ниже.

По умолчанию ваш **Рабочий стол** X не будет позволять X-приложениям подключаться к нему. Можно **разрешить запуск удаленных программ на Рабочем столе** с помощью команды xhost. Используйте ее на своей локальной Linux-системе для контроля над удаленными машинами, которые могут подключаться к X и выводить приложения на вашем **Рабочем столе**. Рассмотрим примеры использования команды xhost:

```
$ xhost          Вывести список хост-компьютеров, которым разрешен доступ
access control enabled, only authorized clients can connect
$ xhost +        Отключить контроль доступа (опасно)
access control disabled, clients can connect from any host
$ xhost -        Повторно активировать контроль доступа
access control enabled, only authorized clients can connect
$ xhost remotemachine Добавить хост-компьютер, которому разрешен доступ
remotemachine being added to access control list
```

Контроль доступа отключается только для задач, связанных с устранением неполадок. Однако если для определенного хост-компьютера (в данном случае **remotemachine**) доступ открыт, можно выполнить следующую операцию в интерпретаторе команд удаленной машины, чтобы приложения X с этой машины появлялись на локальном **Рабочем столе** (в данном случае **localmachine**):

```
$ export DISPLAY=localmachine:0   Установить DISPLAY как localmachine:0
$ xterm &              Открыть удаленный терминал на локальном компьютере
$ xclock &             Открыть удаленные часы на локальном компьютере
$ gtali &              Открыть удаленную игру dice на локальном компьютере
```

После отправки переменной DISPLAY на **remotemachine** для указания на **localmachine** любое приложение, запущенное из интерпретатора команд на удаленном компьютере, отобразится на **Рабочем столе** на локальной машине. В данном случае мы запустили окно терминала, часы и приложение с игрой.

ПРИМЕЧАНИЕ

В последних версиях Ubuntu X-server по умолчанию не ожидает TCP-подключений. Для разрешения удаленных подключений отредактируйте файл /etc/gdm/gdm.conf-custom на X-сервере следующим образом:

```
[security] DisallowTCP=false
```

Затем перезапустите X Windows.

Предоставить таким образом общий доступ к приложениям X между Linux- и UNIX-системами достаточно легко. Однако довольно банально использовать его для связи с другими платформами. Если на вашем компьютере установлена Windows, вам придется запустить X-сервер. Бесплатным решением этой проблемы является программа Cygwin, в которую входит X-сервер. Существуют также многофункциональные коммерческие X-серверы, но они могут быть очень дорогими. Для предоставления общего доступа к удаленным рабочим столам между разными операционными системами мы предлагаем использовать виртуальную сеть передачи данных (VNC).

Предоставление общего доступа к Рабочим столам с помощью VCN

Виртуальная сеть передачи данных состоит из программ для сервера и клиента, которые позволяют получить **доступ к полному отображению Рабочего стола с одного компьютера на другой**. В Ubuntu по умолчанию установлена программа vncviewer для доступа к удаленному рабочему столу с вашего компьютера (клиент), но необходимо установить пакет программ vncserver, чтобы иметь возможность предоставить доступ на ваш Рабочий стол (сервер). Для этого наберите следующее:

```
$ sudo apt-get install vncserver
```

Клиенты и серверы VNC работают со множеством операционных систем. VNC-серверы можно применять в системах Linux, Windows (32-bit), Mac OS X и UNIX. VCN-клиенты используются в вышеперечисленных, а также во множестве других систем (включая OS/2, PalmOS) и даже могут применяться как Java-приложение в браузере.

Настройка VNC-сервера

Предположим, что в своей системе Linux вы используете Рабочий стол, заданный по умолчанию (`DISPLAY=:0`) в качестве локального Рабочего стола. Для начала создадим независимые Рабочие столы, доступные через VNC. С правами суперпользователя в любом текстовом редакторе откройте файл `/etc/vnc.conf` в своей Linux-системе, играющей роль VNC-сервера:

```
# vi /etc/vnc.conf
```

Проверьте настройки в данном файле. Обратите внимание, что файл конфигурации используется каждый раз, когда вы запускаете программу vncserver.

Затем от лица каждого пользователя, которым необходимо будет подключаться к их личным Рабочим столам на VNC-сервере, запустите команду `vncpasswd`. В нашем примере мы запускаем ее от лица пользователя francois:

```
$ vncpasswd
Password: *******
Verify: *******
```

Наконец, вы можете запускать VNC-сервер (vncserver). Наберите следующее от лица пользователя root:

```
$ vncserver
```

ПРИМЕЧАНИЕ

vncserver не установлен как системный процесс по умолчанию. Для получения более подробной информации о назначении системных процессов см. гл. 11.

Если вы используете встроенный в систему брандмауэр iptables, убедитесь, что порты для VNC открыты. Каждый экран работает с отдельным портом. Экран с но-

мером N доступен через TCP-порт 5900+N. Например, экран 1 работает через порт 5901. Обращайтесь к гл. 14 для более подробной информации об iptables.

Запуск VNC-клиента

Как только VNC-сервер запущен, вы можете подключиться к Рабочему столу этого компьютера из любой клиентской системы, упомянутой выше (Windows, Linux, Mac OS X, UNIX и пр.). Предположим, что ваш VNC-сервер находится на компьютере по имени myserver. Вы можете набрать следующую команду, чтобы **запустить клиент** с другой Linux-системы:

```
$ vncviewer myserver:1      Подключиться как francois на экран 1
$ vncviewer myserver:2      Подключиться как chris на экран 2
```

Если вы предварительно не определяли никаких команд для запуска, то увидите фоновый экран окна системы X Window System. Чтобы пойти далее, необходимо запустить программы на системе сервера или с вашего клиента, которые появятся на экране VNC X. Например:

```
$ xterm -display myserver:1 &
$ metacity --display myserver:1 &
```

ПРИМЕЧАНИЕ

В большинстве программ X Window необходимо указать, какой сервер используется (в этом случае сервер VNC) с помощью параметра -display. Однако менеджер окон metacity требует два дефиса в этом параметре --display.

Кроме того, можно использовать tsclient для подключения; в этом примере вы бы просто указали `myserver:1` как компьютер, а VNC — как протокол.

Использование VNC вместе с SSH в ненадежных сетях

VNC считается небезопасным протоколом. Пароль посылается при довольно слабом методе шифрования, а остальная сессия вообще не шифруется. По этой причине при использовании VNC в ненадежных сетях либо в Интернете рекомендуется туннелировать его через SSH.

Работа SSH описывается в разд. «Использование команды ssh для удаленного входа в систему» этой главы. Для переадресации VNC-экрана 2 (порт 5902) на компьютер по имени myserver на тот же локальный порт наберите следующее:

```
$ ssh -L 5902:localhost:5902 myserver
```

ПРИМЕЧАНИЕ

Если вы будете использовать VNC на постоянной основе, взгляните на tightvnc (пакет программ с таким же именем). Хотя он и не включен в Ubuntu, это еще одно воплощение протокола VNC с открытым исходным кодом, которое подвергается активным разработкам. У пакета есть масса новых функций и оптимизаций. Эти функции также включают в себя встроенное ssh-туннелирование.

Пакет программ Vino

Если вы работаете с GNOME и хотели бы предоставить общий доступ к существующему **Рабочему столу** GNOME (`display :0`), то можете сделать это с помощью пакета программ Vino. На рабочей панели GNOME выберите **System ▸ Preference ▸ Remote Desktop** (Система ▸ Настройка ▸ Удаленный рабочий стол) для отображения окна **Remote Desktop Preferences** (Настройки удаленного рабочего стола) (команда `vino-preferences`), показанного на рис. 13.3.

В окне **Remote Desktop Preferences** (Настройки удаленного рабочего стола) можно установить флажок **Allow other users to view your desktop** (Разрешить другим пользователям видеть ваш рабочий стол) и таким образом позволить удаленным пользователям VNC просматривать ваш **Рабочий стол**. Установка флажка **Allow other users to control your desktop** (Разрешить другим пользователям контролировать ваш рабочий стол) позволит пользователям управлять вашим **Рабочим столом** с помощью мыши и клавиатуры.

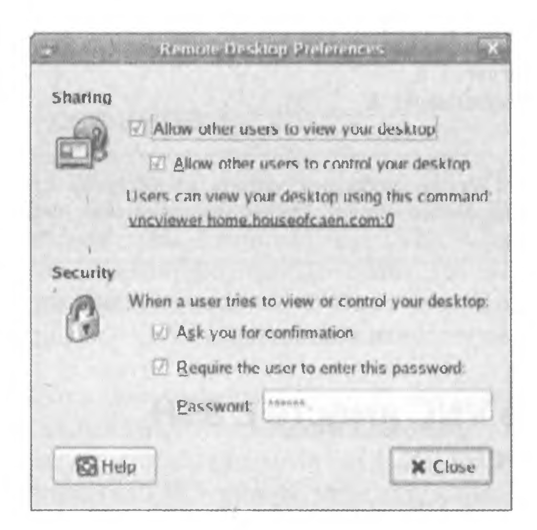


Рис. 13.3. Vino позволяет удаленным пользователям просматривать ваш Рабочий стол и даже управлять им

Если стоит флажок **Ask you for confirmation** (Спрашивать меня о подтверждении), то удаленный запрос на просмотр вашего **Рабочего стола** выведет всплывающее окно для подтверждения соединения до того, как удаленный пользователь сможет просматривать ваш **Рабочий стол**. Установка флажка **Require the user to enter this password** (Требовать от пользователя ввода пароля) — отличная идея. Это не даст тем пользователям, у которых нет пароля, просматривать **Рабочий стол**. Не забудьте проверить, чтобы длина пароля была не менее восьми символов.

Как указано в окне **Remote Desktop Preferences** (Настройки удаленного рабочего стола), вы можете использовать vncviewer из другой системы Linux (с показанным адресом и номером экрана) для отображения общего **Рабочего стола**.

Резюме

Если вам когда-нибудь придется администрировать несколько систем, можете не беспокоиться — Linux предлагает богатый выбор команд для удаленного администрирования. SSH предоставляет шифрованную коммуникацию для удаленного входа в систему, туннелирования и передачи файлов.

Виртуальная сеть передачи данных (VNC) позволяет системе Linux предоставлять клиентской системе общий доступ к **Рабочему столу**, чтобы он появлялся прямо на экране компьютера-клиента. С помощью утилиты Vino можно предоставлять такой вид общего доступа, когда VNC-сервер и клиент могут работать на одном **Рабочем столе** одновременно.

14 Повышение уровня безопасности

Обеспечение безопасности Linux-системы означает, во-первых, ограничение доступа к учетным записям пользователей и сервисам в этой системе. Во-вторых, необходимо удостовериться, что никто не проник за установленные барьеры безопасности.

Ubuntu, Debian и другие системы на основе этих дистрибутивов имеют высокий уровень безопасности по умолчанию. Это означает, что не существует учетных записей пользователей без пароля и большинство сетевых служб (Web, FTP и т. д.) отключено по умолчанию (даже если связанные с ними программы установлены).

Хотя множество команд, описанных в этой книге, могут быть использованы для проверки и повышения уровня безопасности вашей системы, некоторые функции Linux оснащены специальными командами подобного действия. Например, безопасные учетные записи пользователей с хорошей парольной защитой, достойный брандмауэр и единообразный вход в систему (а также его мониторинг) крайне важны для безопасности системы Linux. В этой главе мы описываем команды, относящиеся к этим функциям, и некоторые продвинутые функции, например SELinux и tripwire.

Работа с пользователями и группами

Во время большинства процедур установки Linux вас просят ввести пароль для суперпользователя root (для администрирования системы). Далее, возможно, вас попросят создать учетную запись с выбранным именем и ввести для нее пароль (для ежедневного пользования компьютером). Мы рекомендуем всегда входить в систему в качестве обычного пользователя и только при необходимости работать с командами su или sudo для использования прав суперпользователя. Как только Linux установлена, вы можете применять команды или графические утилиты для создания пользователей, редактирования учетных записей, а также задания и изменения паролей.

Ubuntu повышает безопасность, запрещая вход в систему как суперпользователь по умолчанию. Вместо этого во время установки вы создаете пользователя с паролем, который может выполнять некоторые административные функции. Используйте команду sudo в Ubuntu, чтобы выполнять одиночные root-команды.

Команда sudo запрашивает пароль администратора, обычно являющийся вашим паролем. Это позволит избежать большинства случаев выполнения команд от лица суперпользователя, которые вы не намеревались вводить.

Управление пользователями из графической оболочки

На **Рабочем столе** Ubuntu, поддерживающем приложения X, можно управлять пользователями и группами из окна **User Manager** (Диспетчер пользователей) (**System ▸ Administration ▸ Users and Groups** (Система ▸ Администрирование ▸ Пользователи и группы)). Существует вариант использования графического веб-интерфейса для управления учетными записями пользователей на сервере. Наиболее популярной утилитой такого плана является Webmin (www.webmin.com). Убедитесь, что Webmin не работает через порт, используемый по умолчанию (1000), по причинам, связанным с безопасностью. Вы также можете использовать специальные веб-интерфейсы. Например, существует множество автоматических графических интерфейсов для организации веб-хостинга, таких как cPanel (www.cpanel.com), Plesk (www.swsoft.com/plesk) и Ensim (www.ensim.com).

Добавление учетных записей пользователей

Для **добавления новых пользователей** можно воспользоваться командой useradd. Единственным необходимым аргументом является имя создаваемого пользователя. Вы можете увидеть некоторые установки, используемые по умолчанию, для добавления новых пользователей с помощью параметра -D:

```
$ useradd -D           Показать значения, используемые по умолчанию командой useradd
GROUP=100              Установить идентификатор группы 100 (пользователи)
HOME=/home             Установить корневой каталог /home
INACTIVE=-1            Отключить истечение времени действия пароля (-1)
EXPIRE=                Не устанавливать дату истечения срока действия учетной записи
SHELL=/bin/sh          Установить интерпретатор команд,
                       используемый по умолчанию как /bin/bash
SKEL=/etc/skel         Скопировать файлы конфигурации,
                       используемые по умолчанию из /etc/skel в $HOME
CREATE_MAIL_SPOOL=no   Создать папку для хранения
                       временных данных электронной почты
```

Ubuntu и другие Debian-системы заменяют группу, используемую по умолчанию (100), и создают новую для каждого пользователя. По умолчанию идентификатор (ID) первого созданного пользователя и группы равен 1000. Имя группы совпадает с именем пользователя. Базовый каталог — это имя пользователя, добавленное к /home. Так, например, можно создать первую обычную учетную запись пользователя в системе следующим образом:

```
$ sudo useradd -m willz
```

Результатом станет новая учетная запись пользователя с именем willz (UID 1001) и группа willz (GID 1001). Параметр -m указывает на то, что необходимо

создать домашний каталог /home/willz и копию набора файлов конфигурации (каждый из них начинается с точки (.)) в домашний каталог из папки /etc/skel. Учетная запись будет активной неопределенное время (без даты истечения срока действия). **Добавьте пароль** следующим образом.

```
$ sudo passwd horatio
Changing password for user horatio
New UNIX password: ********
Retype new UNIX password: ********
passwd: all authentication tokens updated successfully.
```

В большинстве случаев это все, что необходимо сделать, чтобы иметь рабочую учетную запись пользователя.

ПРИМЕЧАНИЕ

Помните, что надо использовать сложные пароли.

Существует множество параметров для **замены используемых по умолчанию параметров** во время создания нового пользователя. Совмещайте различные параметры по своему выбору. Рассмотрим несколько примеров:

```
$ sudo useradd -u 1101 -g 1300 skolmes   Использовать заданный UID
                                         (идентификатор пользователя) и GID
                                         (идентификатор группы) для пользователя
$ sudo useradd -m -d /home/jj jones      Создать корневой каталог /var/x/jj
$ sudo useradd -G support,sales timd     Добавить пользователя в группы
                                         support и sales
$ sudo useradd -c "Tom G. Lotto" tlot    Добавить полное имя пользователя
                                         в поле комментария
$ sudo useradd -s /bin/tcsh joeq         Назначить новый интерпретатор
                                         команд, используемый по умолчанию
                                         (tcsh); вы должны его установить
$ sudo useradd -e 2008-04-01 jerry       Дата истечения срока действия
                                         учетной записи April 01, 2008
$ sudo useradd -f 0 jdoe                 Создать отключенную учетную запись
$ sudo useradd -s /sbin/nologin billt    Не позволять пользователю входить
                                         в систему
$ sudo useradd billyq                    Предотвратить создание корневого
                                         каталога, отсутствие -m
```

Перед тем как добавить пользователя в группу, нужно создать эту группу (см. команду groupadd, описываемую в подразд. «Добавление групп» далее в этом разделе). Пользователь должен принадлежать к изначальной группе, которую можно определить с помощью параметра -g, а также к дополнительным группам, что определяется параметром -G.

Для **вывода списка групп, к которым принадлежит пользователь**, используйте команду groups:

```
$ groups francois    Вывод списка групп, к которым принадлежит пользователь
francois ftpusers
```

Пример с созданием учетной записи со сроком действия (-e) является довольно полезным для установки даты окончания действия учетной записи временного

пользователя. Измените интерпретатор команд, используемый по умолчанию, на nologin, если хотите, чтобы пользователь имел доступ к компьютеру (через FTP, POP3 и т. д.), но не мог работать с обычным интерпретатором команд Linux. Аналогично, в последнем примере, где для создания корневого каталога не применяется параметр -m, пользователю позволяется иметь доступ к компьютеру, однако у него не будет базовой директории. Обратите внимание, что, если вы не добавите параметр -m, команда useradd не создаст корневого каталога для пользователя.

Изменение настроек, используемых по умолчанию командой useradd. Значения, используемые по умолчанию во время создания новой учетной записи пользователя с помощью useradd (интерпретатор команд по умолчанию, GID, даты истечения срока действия и т. д.), устанавливаются в файлах /etc/login.defs и /etc/default/useradd. Вы можете отредактировать эти файлы для изменения значений, используемых по умолчанию, либо запустить команду useradd с параметром -D для вывода списка или выборочного изменения значений:

```
$ useradd -D    Вывести значения, используемые по умолчанию для useradd
$ sudo useradd -D -b /home2 -s /bin/csh  Установить базовую директорию
                                         и интерпретатор команд, используемый
                                         по умолчанию
$ sudo useradd -D -e 2009-01-01          Установить дату истечения срока действия
                                         всех учетных записей как 2009-й год
```

Как указано выше, после создания учетной записи пользователя с параметром -m файлы и папки из /etc/skel копируются в корневой каталог нового пользователя. Они включают в себя некоторые файлы интерпретатора команд bash и ссылку на папку-образец. Вы можете добавить другие файлы и папки в /etc/skel, чтобы все новые пользователи получали их. Например, занимаясь настройкой веб-сервера, вы можете создать папки public_ftp и public_html, чтобы пользователи могли добавлять туда страницы и файлы для общего доступа.

Изменение учетных записей пользователей

После того как учетная запись пользователя создана, можно **изменить ее настройки с помощью команды usermod**. Большинство параметров совпадают с используемыми командой useradd. Например:

```
$ sudo usermod -c "Thomas Lotto" tlot   Изменить имя пользователя в поле
                                        комментария
$ sudo usermod -s /bin/sh joeq          Изменить интерпретатор команд,
                                        используемый по умолчанию на sh
$ sudo usermod -L swanson               Заблокировать учетную запись
                                        пользователя swanson
$ sudo usermod -U travis                Разблокировать учетную запись
                                        пользователя travis
```

Обратите внимание, что два последних примера соответственно блокируют и разблокируют учетную запись пользователя. Блокировка пользователя не удаляет из системы учетную запись, пользовательские файлы и папки. Однако она предотвращает вход этого пользователя в систему. Блокировка учетной записи полезна, когда работник покидает компанию, однако его работа должна быть

передана другому человеку. При таких обстоятельствах блокировка, а не удаление записи предотвращает ситуацию, когда файлы, принадлежавшие пользователю, оказываются принадлежащими несуществующему идентификационному номеру.

Поскольку обычный пользователь не может применять команды useradd и usermod, существуют специальные команды для **изменения личной информации в учетной записи**. Вот несколько примеров:

```
$ chsh -s /bin/sh                  Изменить текущий интерпретатор команд на /bin/sh
$ sudo chsh -s /bin/sh francois    Изменить интерпретатор команд на /bin/sh
$ sudo chfn \
   -o "B-205"          \           Изменить номер офиса
   -h "212-555-1212"   \           Изменить домашний номер телефона
   -w "212-555-1957"               Изменить офисный номер телефона
$ finger francois
Login: francois                              Name: Francois Caen
Directory: /home/francois                    Shell: /bin/bash
Office: B-205, 212-555-1212                  Home Phone: 212-555-1957
On since Sat Aug 4 13:39 (CDT) on tty1     4 seconds idle
No mail.
No Plan.
```

Информация, измененная командой chfn и отображенная с помощью finger, хранится в пятом поле файла etc/passwd для выбранного пользователя (этот файл может редактироваться только пользователем root, и выполнять данную операцию нужно с большой осторожностью, применяя команду vipw).

В других версиях Linux можно использовать параметр -f с командой chfn для изменения вашего реального или полного имени. В Ubuntu эта функция отключена по умолчанию. Вы можете изменить данное обстоятельство, отредактировав файл /etc/login.defs. Найдите данную строку:

```
CHFN_RESTRICT         rwh
```

и измените ее на такую:

```
CHFN_RESTRICT         frwh
```

Удаление учетных записей пользователей

Посредством команды userdel можно **удалять учетные записи пользователей** из системы, а также другие файлы (корневые каталоги, спул-файлы электронной почты и т. д.). Рассмотрим примеры:

```
# userdel jimbo      Удалить пользователя, но не его корневой каталог
# userdel -r lily    Удалить пользователя, корневой каталог
                     и спул-файлы электронной почты
```

Управление паролями

Добавить или изменить пароль легко с помощью команды passwd. Однако у нее существуют дополнительные параметры, позволяющие администраторам управлять такими вещами, как блокировка учетных записей, истечение срока действия

паролей и сообщения о необходимости смены пароля. Помимо passwd, для работы с паролями предназначены команды chage, chfn и vipw.

Обычные пользователи могут **изменять только свой пароль**, в то время как суперпользователь имеет право менять пароли любого пользователя. Например:

```
$ passwd                Изменить личный пароль пользователя
Changing password for user chris.
Changing password for chris. (current) UNIX password: ********
New UNIX password: *
BAD PASSWORD: it's WAY too short
New UNIX password: *********
Retype new UNIX password: *********
passwd: password updated successfully
$ sudo passwd joseph    Root-пользователь может менять
                        пароль любого пользователя
Changing password for user joseph.
New UNIX password: *
Retype new UNIX password: *
passwd: password updated successfully
```

В первом примере обычный пользователь (chris) меняет свой собственный пароль. Даже будучи зарегистрированным в системе, он должен ввести свой текущий пароль перед вводом нового. К тому же команда passwd предотвращает установку слишком короткого пароля, основанного на словарном слове, с недостатком разнообразных символов или такого пароля, который легко угадать. Во втором примере root-пользователь может изменить пароль любого пользователя без старого пароля.

Пароли должны иметь как минимум восемь символов, одновременно состоять из букв и других символов (цифр, знаков препинания и т. д.), а также не должны содержать реальных слов. Создайте легкий для запоминания пароль, но чтобы его было сложно угадать.

Системный администратор может использовать команду passwd **для блокировки и разблокировки учетных записей** пользователей. Например:

```
$ sudo passwd -l carl       Блокировать учетную запись пользователя (carl)
Locking password for user carl. passwd: Success
$ sudo passwd -u carl       Разблокировать учетную запись пользователя (carl)
Unlocking password for user carl. passwd: Success
$ sudo passwd -u jordan     Неудачная попытка разблокировать учетную запись
                            с пустым паролем
Unlocking password for user jordan.
passwd: Warning: unlocked password would be empty.
passwd: Unsafe operation (use -f to force)
```

Блокировка учетной записи с помощью команды passwd подставляет восклицательный знак (!) перед полем с паролем в файле /etc/shadow (где хранятся пользовательские пароли). Когда учетную запись разблокируют, восклицательный знак удаляется и восстанавливается предыдущий пароль пользователя.

Администратор может использовать команду passwd, чтобы **обязать пользователей регулярно менять пароль**, а также для того, чтобы **оповестить о том, что**

срок действия пароля истекает. Для использования функции истечения срока действия пароля в учетной записи пользователя должно быть активировано истечение срока действия пароля. Следующие примеры используют команду passwd для работы с истечением срока действия пароля:

```
$ sudo passwd -n 2 vern      Установить минимальный срок действия
                             пароля 2 дня
$ sudo passwd -x 300 vern    Установить максимальный срок действия пароля
                             300 дней
$ sudo passwd -w 10 vern     Оповещать об истечении срока действия
                             пароля за 10 дней
$ sudo passwd -i 14 vern     Количество дней до отключения учетной записи
                             после истечения срока действия пароля
```

В первом примере пользователь должен подождать хотя бы два дня (-n 2) до установки нового пароля. Во втором пользователь должен поменять свой пароль в течение 300 дней (-x 300). В следующем примере пользователя предупреждают за 10 дней до истечения срока действия пароля (-w 10). В последнем примере учетная запись пользователя отключается через 14 дней после истечения срока действия пароля (-i 14).

Для **просмотра сроков действия паролей** можно воспользоваться командой chage:

```
$ sudo chage -l vern  Показать информацию о дате истечения пароля
Last password change                                  : Aug 04, 2007
Password expires                                      : May 31, 2008
Password inactive                                     : Jun 14, 2008
Account expires                                       : never
Minimum number of days between password change        : 2
Maximum number of days between password change        : 300
Number of days of warning before password expires     : 10
```

Как системный администратор вы можете использовать команду chage для управления сроками истечения действия паролей. Кроме возможности установки минимального (-m) и максимального (-M) количества дней действия, а также дней до выдачи предупреждения (-W), команда chage позволяет указать день, в который пользователю необходимо задать новый пароль, или определенную дату, когда учетная запись становится неактивной:

```
$ sudo chage -I 40 frank     Отключить учетную запись пользователя
                             по прошествии 40 дней
$ sudo chage -d 5 perry      Срок действия пароля закончится через 5 дней
```

Вместо пяти дней (-d 5) вы можете установить для этого параметра значение 0, что вынудит пользователя изменить свой пароль в следующий раз при входе в систему. Например, при установке -d 0 у пользователя **perry** запросили бы новый пароль во время следующего входа в систему:

```
login: perry
Password: ********
You are required to change your password immediately (root enforced)
Changing password for perry.
```

```
(current) UNIX password:
New UNIX password: *********
Retype new UNIX password: *********
```

Добавление групп

Каждый новый пользователь приписан к одной или нескольким группам. Вы можете создавать группы в любое время и добавлять в них пользователей. Права каждой группы на использование файлов и папок в Linux зависят от того, как распределены права доступа. Добавление пользователя в группу позволяет определить права доступа к файлам, каталогам и приложениям. Таким образом, пользователи могут работать вместе над проектом или иметь общий доступ к ресурсам.

Команды, подобные тем, что применяются для работы с пользователями, доступны и для управления группами. Вы можете добавлять (groupadd) и удалять группы (groupdel), изменять их настройки (groupmod), а также добавлять и удалять членов этих групп (groupmems). Рассмотрим несколько примеров **создания новых групп** с помощью команды groupadd:

```
$ sudo groupadd marketing          Создать новую группу со следующим доступным GID
$ sudo groupadd -g 1701 sales      Создать новую группу с GID, равным 1701
$ sudo groupadd -o -g 74 mysshd    Создать группу с существующим GID
```

Применяя команду groupmod, вы можете **изменять имя или идентификационный номер (ID)** существующей группы. Например:

```
$ sudo groupmod -g 491 myadmin         Изменить GID группы myadmin на 491
$ sudo groupmod -n myad myadmin        Изменить имя группы myadmin на myad
```

Для того чтобы удалить существующую группу, используйте команду groupdel:

```
$ sudo groupdel myad  Удалить существующую группу myad
```

Обратите внимание, что удаление группы или пользователя не удаляет файлы, папки, устройства или другие элементы, принадлежащие группе или пользователю. Если вы выведете полный список (ls -l) файлов или папок, приписанных удаленной группе или пользователю, то будут показаны UID и GID удаленного пользователя или группы.

Наблюдение за пользователями

Создав учетные записи, которые позволяют пользователям работать за вашим компьютером, можете понаблюдать за их активностью. Для этого предназначены специальные команды, которые уже были описаны в предыдущих главах.

- Используйте команду find (см. гл. 4) для поиска файлов по всей системе, которые принадлежат определенному пользователю.
- Применяйте команду du (см. гл. 7), чтобы увидеть, сколько места занимают папки выбранных пользователей.
- Используйте команды fuser, ps и top (см. гл. 9) для просмотра запущенных пользователями процессов.

Помимо умомянутых выше, существуют команды для просмотра всех зарегистрированных в системе пользователей на данный момент, а также для получения общей информации о пользователях с учетными записями в вашей системе. Рассмотрим команды для **получения информации о пользователях, находящихся в системе**:

```
$ last          Вывести список недавних удачных входов в систему
greek    tty3                Sun Aug     5 18:05     still logged in
chris    tty1                Sun Aug     4 13:39     still logged in
root     pts/4 thompson      Sun Aug     5 14:02     still logged in
chris    pts/1 :0.0          Sat Aug     4 15:47     still logged in
3 13:46 - 15:40 (01:53)
francois pts/2               Thu Aug 2 11:14 - 13:38 (2+02:24)
$ last -a       Упрощает процесс чтения имени хост-компьютера удаленного клиента
$ sudo lastb    Вывести список недавних неудачных входов в систему
julian   ssh:notty   ritchie           Mon Aug 6 12:28 - 12:28    (00:00)
morris   ssh:notty   thompson          Tue Jul 31 13:08 - 13:08   (00:00)
baboon   ssh:notty   10.0.0.50         Sun Jul  8 09:40 - 09:40   (00:00)
francois ssh:notty   000db9034dce.cli  Fri Jun 22 17:23 - 17:23   (00:00)
$ who -u        Вывести список всех пользователей в системе
                на данный момент (полный вариант)
greek           tty3  2007-08-05 18:05 17:24        18121
jim             pts/0 2007-08-06 12:29              20959 (server1.example.com)
root            pts/3 2007-08-04 18:18 13:46   17982 (server2.example.com)
francois pts/2 2007-07-31 23:05        old    4700 (0a0d9b34x.example.com)
chris           pts/1 2007-08-04 15:47         old    17502 (:0.0)
$ users         Вывести список всех пользователей в системе
                на данный момент (краткий вариант)
chris francois greek jim root
```

Применяя команду last, вы можете увидеть каждого зарегистрировавшегося в системе пользователя (или каждого открывшего новый интерпретатор команд) либо узнать, как долго они находились в системе. Вы также можете увидеть отметку "still logged in" (все еще в системе). Строки терминала tty1 и tty3 показывают пользователей, работающих с виртуальных терминалов в консоле. Строки pts означают людей, открывших интерпретатор команд с удаленного компьютера (thompson) или локальный экран X (:0.0). Мы рекомендуем использовать параметр -a для улучшенного отображения данных. Команда lastb показывает неудачные попытки входа в систему и их направление. Команды who -u и users показывают информацию о текущих пользователях, находящихся в системе.

Рассмотрим несколько команд для **получения дополнительной информации о находящихся в системе пользователях**:

```
$ id            Ваша принадлежность (UID, GID и группа
                для текущего интерпретатора команд)
uid=1000(chris) gid=1000(chris) groups=4(adm),20(dialout),24(cdrom),
25(floppy),29(audio),30(dip),44(video),46(plugdev),104(scanner),112(netdev),
113(lpadmin),115(powerdev),117(admin),1000(chris)
$ who am i            Ваша принадлежность (пользователь, tty,
                      дата входа в систему, местонахождение)
chris    pts/0 Aug 3 2140 (:0.0)
```

```
$ finger -s chris      Информация о пользователе (краткая)
Login    Name          Tty         Idle        Login Time    OfficeOffice Phone
chris    Chris Negus   tty1        1d          Aug 4 13:39   A-111 555-1212
$ finger -l chris      Информация о пользователе (полная)
Login: chris                                   Name: Chris Negus
Directory: /home/chris                         Shell: /bin/bash
Office: A-111, 555-1212                        Home Phone: 555-2323
On since Sat Aug 4 13:39 (CDT) on tty1         2 days idle
New mail received Mon Aug 6 13:46 2007 (CDT)
    Unread since Sat Aug 4 09:32 2007 (CDT)
No Plan.
```

Помимо вывода стандартной информации о пользователе (имя, корневой каталог, интерпретатор команд и т. д.), команда finger также отображает информацию, хранящуюся в специальных файлах в корневом каталоге пользователя. Например, содержание файлов пользователя ~/.plan и ~/.project (если они существуют) отображается в конце вывода команды finger. С существующим файлом .project в одну строку и файлом .plan в несколько строк это может выглядеть следующим образом:

```
$ finger -l chris      Информация о пользователе (полная информация,
                       файлы .project и .plan)
...
Project:
My project is to take over the world. Plan:
My grand plan is
to take over the world
by installing Linux on every computer
```

Настройка встроенного брандмауэра

Брандмауэр — одна из важнейших утилит для поддержания компьютера в безопасности, исключающая возможность вторжения в систему через сеть или Интернет. Он может защищать компьютер, проверяя каждый пакет с данными, поступающими в сетевые интерфейсы, а затем принимая решение о дальнейшей судьбе пакета в соответствии с заданными вами параметрами. Брандмауэр, встроенный в текущее ядро Linux, называется iptables (вы также могли слышать о ipchains — предшественнике iptables в ядрах версии 2.2 и ниже). В Ubuntu iptables запускается и настраивается в процессе установки.

Функция iptables (**www.netfilter.org**) является невероятно мощной, но достаточно сложной для использования через командную строку. По этой причине многие устанавливают основные правила поведения брандмауэра посредством графического интерфейса. Чтобы получить графический интерфейс, установите пакет программ firestarter. В нем есть помощник для настройки брандмауэра. Для запуска Firestarter выберите **System ▸ Administration ▸ Firestarter** (Система ▸ Администрирование ▸ Firestarter). Вы также можете опробовать такие дополнительные пакеты программ, как FWBuilder (пакет программ fwbuilder) и Shorewall (пакет программ shorewall) для графической настройки брандмауэров.

Устанавливая Ubuntu, вы одновременно установили в систему брандмауэр. Ubuntu создает конфигурацию iptables, которая является хорошим началом для использования брандмауэра через **Рабочий стол**. Эта конфигурация открывает только несколько портов для работающих демонов и блокирует остальные. Вы можете изменить настройки по умолчанию с помощью команд, описанных далее.

ПРИМЕЧАНИЕ

Прежде чем читать дальше, просмотрите документ для Ubuntu под названием IpTables HowTo, расположенный на сайте https://help.ubuntu.com/community/IptablesHowTo. В нем содержится много полезной информации об использовании iptables в Ubuntu, сильно отличающемся от других версий Linux, таких как Fedora.

Для более сложных задач, когда, к примеру, iptables используется как брандмауэр нескольких компьютеров, мы рекомендуем воспользоваться одной из графических утилит, упомянутых выше. Однако бывают случаи, когда у вас нет доступа к графическому интерфейсу или вам необходимо воспользоваться функцией, которая недоступна через него. Вот тогда знание синтаксиса команды `iptables` оказывается полезным для вывода списка текущих правил и самостоятельного добавления новых.

Перед тем как вы начнете работать с брандмауэром в Ubuntu, нужно проверить, как он настроен в вашей системе. Рассмотрим способ **выведения списка текущих правил** брандмауэра в системе Linux:

```
$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target          prot opt source           destination
Chain FORWARD (policy ACCEPT)
target          prot opt source           destination
Chain OUTPUT (policy  ACCEPT)
target          prot opt source           destination
```

После установки Ubuntu вы увидите вывод предыдущей команды, означающий, что ни одно правило не было задано. После некоторых изменений вы можете увидеть уже созданный список правил, подобный тому, что показан ниже:

```
$ sudo iptables -L         Отобразить текущую таблицу фильтров iptables
Chain INPUT (policy ACCEPT)
target          prot opt source           destination
...
ACCEPT    udp    --    anywhere    anywhere    udp dpt:ipp
ACCEPT    tcp    --    anywhere    anywhere    tcp dpt:ipp
ACCEPT    0      --    anywhere    anywhere    state RELATED,ESTABLISHED
ACCEPT    tcp    --    anywhere    anywhere    state NEW tcp dpt:ftp
ACCEPT    tcp    --    anywhere    anywhere    state NEW tcp dpt:ssh
ACCEPT    tcp    --    anywhere    anywhere    state NEW tcp dpt:http
REJECT    0      --    anywhere    anywhere    reject-with icmp-host-prohibited
Chain FORWARD (policy  ACCEPT)
target          prot opt source           destination
REJECT    0      --    anywhere    anywhere    reject-with icmp-host-prohibited

Chain OUTPUT (policy ACCEPT)
target          prot opt source           destination
```

Пример демонстрирует таблицу фильтров брандмауэра iptables, используемую по умолчанию. Она показывает, что из всех пакетов, поступающих в сетевые интерфейсы компьютера, пакеты для протокола печати через Интернет (ipp) по протоколам udp и tcp допускаются в систему. К тому же принимаются tcp-пакеты, направленные в порты FTP (ftp), Secure Shell (ssh) и Web (http). Кроме того, прием пакетов происходит, если они ассоциируются с установкой соединения. Ниже вы можете **ознакомиться с nat-таблицей**:

```
$ sudo iptables -t nat -L          Отобразить текущую nat-таблицу
                                   программы iptables
Chain PREROUTING (policy  ACCEPT)
target         prot opt source          destination
DNAT      tcp   --     0.0.0.0/0    11.22.33.44  tcp dpt:8785 to:10.0.0.155:22
DROP      tcp   --     0.0.0.0/0    0.0.0.0/0    tcp dpt:135
DROP      udp   --     0.0.0.0/0    0.0.0.0/0    udp dpt:135
Chain POSTROUTING (policy ACCEPT)
target         prot opt source          destination
MASQUERADE     all    --     0.0.0.0/0   0.0.0.0/0
Chain OUTPUT (policy ACCEPT)
target         prot opt source          destination
```

Показанная nat-таблица относится к функции преобразования сетевых адресов (Network Address Translation, NAT). NAT позволяет использовать секретные адреса с помощью брандмауэра. Когда пакеты посылаются одним из внутренних компьютеров локальной сети, IP-адрес источника преобразуется с помощью внешнего интерфейса брандмауэра. Брандмауэр следит за этими сессиями, чтобы принимать трафик для машин локальной сети. Все это настраивается в строке MASQUERADE из цепочки POSTROUTING.

В предыдущем примере строка DNAT из цепочки PREROUTING указывает на то, что все запросы на IP-адрес 11.22.33.44 с портом 8785 перенаправляются на внутренний IP-адрес в локальной сети 10.0.0.155 с портом 22 (уловка, позволяющая комулибо подключиться к компьютеру в обход брандмауэра, используя нестандартный порт).

Вот еще несколько примеров, показывающих, как **выводить информацию о брандмауэре**. Команда ipstate является частью пакета программ ipstate.

```
$ sudo iptables -n -L        Правила фильтров, IP-адреса (без просмотра
                             DNS-записей)
$ sudo iptables -v -L        Расширенный вывод (с подсчетом пакетов/бит))
$ sudo iptables -L --line-numbers Показывать номер строки
                                  в цепочке для каждого правила
$ sudo iptables -nvL --line-numbers      Любимое сочетание нашего
                                         технического редактора
```

Вы можете **очистить или остановить все правила iptables** в Ubuntu следующим образом:

```
$ sudo iptables -F    Очистить все правила iptables
```

Данная команда устраняет все правила, поэтому будьте осторожны с ее использованием. Вам придется немедленно установить новые правила.

Рассмотрим несколько примеров того, как использовать команду iptables для **изменения правил активного брандмауэра**:

```
$ sudo iptables -A INPUT -p TCP       \   Добавить правило внутреннего фильтра
    -i eth0                           \   для TCP-пакетов первого Ethernet-интерфейса
    --destination-port 25       \     Направление в порт электронной почты (25)
    -j ACCEPT                         Принимать пакеты
$ sudo iptables -t nat          \     Добавить nat-правило
    -A POSTROUTING              \     Цепочка POSTROUTING
    -o eth1                     \     Пакеты, полученные интерфейсом eth1
    -j SNAT                     \     Перейти к преобразованию сетевых адресов
    --to-source 11.22.33.1            Использовать адрес 11.22.33.1
                                      для отправки исходящей информации
```

Первый пример создает правило, позволяющее принимать новые запросы, входящие в систему через порт 25. Возможной причиной может служить то, что вы настроили компьютер как почтовый сервер (с sendmail, postfix или другим SMTP-сервисом). Второй пример создает nat-правило, которое позволяет брандмауэру использовать преобразование сетевого адреса источника (Source Network Address Translation, SNAT). Благодаря функции SNAT после прохождения информации через брандмауэр создается скрытый IP-адрес, через который можно поддерживать связь с Интернетом, используя внешний IP-адрес брандмауэра.

Для того чтобы иметь возможность применять SNAT или любую другую форму NAT, вам необходимо **разрешить перенаправление IP** на компьютере. Это можно сделать, отредактировав файл /etc/sysctl.conf путем снятия закомментированности следующей переменной:

```
net.ipv4.conf.default.forwarding=1
```

Если вы имеете доступ к компьютеру с выходом в Интернет за границами действия брандмауэра, то можете настроить брандмауэр для **перенаправления пакетов этой службы** на такой компьютер. В следующем примере используется функция под названием *перенаправление портов* для передачи запросов определенной службы через брандмауэр на внешний компьютер назначения:

```
$ sudo iptables -t nat -A PREROUTING \        Добавить nat-правило
                                              для цепочки PREROUTING
    -p tcp -d 11.22.33.1        \     Принимать tcp-запросы на 11.22.33.1
    --dport 80                        \     Для порта 80 (веб-служба)
    -j DNAT                                 \     Перейти к цели DNAT
    --to-destination 10.0.0.2         Перенаправить эти пакеты на 10.0.0.2
```

Можно создать множество других типов правил, изменяющих поведение брандмауэра. Для получения более подробной информации по использованию программы iptables обратитесь к руководству, расположенному на сайте Netfiller (www.netfiller.com).

После внесения описанных изменений вы увидите, что определены следующие правила:

```
$ sudo iptables -t nat -L
Chain PREROUTING (policy ACCEPT)
target    prot  opt   source              destination
```

```
DNAT      tcp   --    anywhere          11.22.33.1   tcp dpt:www to:10.0.0.2
Chain POSTROUTING (policy ACCEPT)
target    prot  opt   source            destination
SNAT      0     --    anywhere          anywhere     to:11.22.33.1
Chain OUTPUT (policy ACCEPT)
target    prot  opt   source            destination
```

Все изменения в правилах iptables сохраняются только на время текущей сессии. После перезагрузки компьютера активным станет набор правил, используемых по умолчанию. Обычно это не то, что нужно. Для **сохранения ваших правил iptables** запустите команду iptables-save:

```
$ sudo iptables-save > iptables.rules    Сохраняет правила в файл
                                         в текущем каталоге
$ sudo cp iptables.rules /etc            Копирует сохраненные правила в /etc
```

Двухфазовый процесс необходим из-за прав доступа к папке /etc (вы можете изменить их, но это не очень хорошая идея). Наконец, правила сохранены для дальнейшего использования.

Далее можно настроить Ubuntu для загрузки этих сохраненных правил для каждого подключенного Ethernet-интерфейса (правила уникальны для каждой сетевой карты или интерфейса в системе). Отредактируйте файл /etc/network/interfaces. После каждой информации о настройках iface для Ethernet-интерфейса, например для eth0, вызывайте команду iptables-restore, как показано в следующем отрывке файла:

```
auto eth0
iface eth0 inet dhcp
   pre-up iptables-restore < /etc/iptables.rules
```

Это дополнение к файлам правил интерфейсов вызывает функцию iptables-restore для восстановления правил, сохраненных ранее в /etc/iptables.rules.

Работа с файлами системного журнала

Большинство Linux-систем настроены на ведение записей о действиях, которые происходят в рамках данных систем. Эти действия затем записываются в файлы журнала, находящиеся в папке /var/log или в ее подпапках. Такие записи производятся программным обеспечением Syslog.

Для управления ведением журнала событий Ubuntu использует syslogd (демон записи системного журнала) и klogd (демон записи журнала ядра) установочных пакетов sysklogd и klogd. Эти демоны включаются автоматически посредством сценария запуска syslog (/etc/init.d/sysklogd). Затем информация об активности системы, основанная на настройках файла /etc/syslog.conf, направляется в файлы папки /var/log, например messages, secure, cron и boot.log.

Периодическое создание файлов журнала обеспечивается программой logrotate с настройками, хранящимися в файле /etc/logrotate.conf и папке /etc/logrotate.d. Cron-процесс команды /etc/cron.daily/logrotate позволяет периодически в течение дня заполнять журнальные файлы.

Вы можете просмотреть любой файл журнала вручную (с помощью vi или другого текстового редактора). Однако, установив пакет logwatch, вы будете каждый день получать наиболее существенную информацию из журнальных файлов на почтовый ящик пользователя root. Можно изменить адреса отправки и получения этого сообщения в файле /etc/cron.daily/0logwatch. Чтобы избежать замкнутых циклов отправки электронной почты, укажите существующий адрес для отправителя, в то время как адрес получателя должен находиться за пределами локального компьютера. Другой способ изменения получателя — это перенаправление электронной почты root-пользователя на другой адрес. Это можно сделать, отредактировав файл /etc/aliases и запустив программу newaliases для вступления изменений в силу. В противном случае просто зайдите в систему как суперпользователь и работайте с почтовой программой пользователя, как это описано в гл. 12, для прочтения электронных сообщений программы logwatch.

Вы можете **отсылать свои собственные сообщения службе ведения системного журнала syslogd** с помощью команды logger. Вот несколько примеров:

```
$ logger Added new video card              Информация добавлена в файл сообщения
$ logger -p info -t CARD -f /tmp/my.txt          Приоритет, тег, файл сообщения
```

В первом примере фраза Added new video card отправляется в файл сообщения. Во втором примере приоритет письма установлен как info и тег CARD добавляется в каждую строку сообщения. Текст сообщения извлекается из файла /tmp/my.txt. Чтобы просматривать системный журнал в реальном времени, используйте команды tail -f или less, как это описано в гл. 5.

Продвинутые утилиты по обеспечению безопасности

На десятке страниц невозможно рассказать обо всем многообразии утилит по обеспечению безопасности, доступных администраторам Linux-систем. Помимо команд, рассмотренных в этой главе, мы представляем описание некоторых достойных внимания функций для дальнейшей зашиты Linux.

- **Security Enhanced Linux (SELinux)** — функция предоставляет средства для защиты файлов, папок и программ Linux-системы, предотвращая несанкционированный доступ из одной используемой системной области в другую. Например, если бы злоумышленники получили незаконный доступ к вашему веб-демону, это не означало бы, что они получили бы доступ к остальной системе. SELinux был разработан Агентством национальной безопасности США, на сайте которого (www.nsa.gov/selinux/info/faq.cfm) есть путеводитель по часто задаваемым вопросам, связанным с данной программой. SELinux устанавливается с помощью нескольких пакетов. Обращайтесь на страницу https://wiki.ubuntu.com/SELinux для получения более подробной информации.
- **Централизованный системный журнал** — если вы управляете несколькими серверами Linux, то для вас предпочтительнее хранить файлы системного журнала на центральном сервере syslog. Используя свой syslog-сервер, вы можете

просматривать информацию о системных событиях с помощью пакета syslog-ng. Кроме того, если пакет logwatch уже не подходит вам из-за недостатка функциональности, попробуйте использовать анализатор системного журнала Splunk.

- **Tripwire** — с помощью этого приложения вы можете сделать моментальный снимок всех файлов в системе, а затем использовать его, чтобы найти изменения в файлах. Эта функция особенно полезна, когда нужно выяснить, подвергались ли определенные программы ненужным изменениям. Вы берете базовые характеристики системного файла, затем регулярно запускаете tripwire и проводите проверку целостности, чтобы увидеть возможные изменения в файлах конфигурации или приложениях.
- **База данных APT** — еще одним способом проверить, подвергались ли программы изменениям, является использование команд APT для проверки достоверности приложений и файлов конфигурации, установленных в системе. Обращайтесь к гл. 2 для получения информации об использовании команд apt и dpkg для проверки содержимого установленных пакетов программ.
- **Chkrootkit** — если вы подозреваете, что в систему проникли злоумышленники, скачайте и соберите программу chkrootkit с сайта www.chkrootkit.org. Это поможет обнаружить руткиты[1], которые могли быть использованы для захвата компьютера. Мы рекомендуем запускать chkrootkit с LiveCD или после монтирования подозрительного диска в чистой системе.

Резюме

В то время как существует множество утилит для защиты Linux, первая линия обеспечения безопасности начинается с защиты учетных записей пользователей и служб, работающих в системе. Команды useradd, groupadd и password являются стандартными средствами для настройки учетных записей пользователей и групп.

Поскольку большинство взломов, происходящих извне, осуществляются злоумышленниками, получающими доступ к системам через общедоступные сети, то настройка брандмауэров важна для любой системы, подключенной к Интернету. Утилита iptables предоставляет функции брандмауэра, встроенные в ядро Linux.

Программа Syslog следит за процессами, происходящими в системе. Она записывает информацию практически обо всех действиях. С помощью пакетов программ logrotate и logwatch, установленных по умолчанию, можно легко управлять файлами системного журнала и ежедневно проверять их.

[1] Набор утилит, которые хакер устанавливает на взломанном компьютере после получения первоначального доступа.

Приложение 1. Использование редакторов vi и vim

Хотя легкие в использовании графические редакторы (такие как gedit и kedit) всегда доступны в Linux, большинство опытных пользователей все еще работают в vi или Emacs при редактировании текстовых файлов. Помимо того, что vi и Emacs работают из любого интерпретатора команд (графический интерфейс необязателен), они имеют еще несколько преимуществ, например возможность работы в редакторе только с помощью клавиатуры и интеграция с удобными утилитами. Кроме того, в отличие от графических текстовых редакторов, с редакторами, основанными на тексте, можно работать, имея подключение к Интернету на малой скорости, например коммутируемый доступ или подключение через спутник.

Это приложение рассматривает функции редактора vi для базового редактирования текста, но также поможет вам научиться более сложным операциям с текстом. Мы предпочли рассказать о vi, а не о Emacs, так как он более универсальный и гибкий, а также потому, что горячие клавиши vi можно нажимать двумя руками. Поскольку многие системы Linux вместо старого vi используют редактор vim (Vi Improved — «Vi Улучшенный»), примеры в этом приложении касаются также и vim. Вот некоторые функции, которые есть в vim, но отсутствуют в vi: множественные уровни отмены ввода, выделение синтаксиса и помощь в режиме онлайн.

ПРИМЕЧАНИЕ

Если вы никогда прежде не работали с vi или vim, обратите внимание на инструкцию, входящую в расширенный пакет программ vim. Запустите команду tutor и следуйте инструкциям для ознакомления со многими ключевыми функциями этих редакторов.

Начало работы

Если вы хотите потренироваться в использовании vi, то сначала скопируйте текстовый файл. Например, введите:

```
$ cp /etc/passwd /tmp
```

Затем откройте этот файл с помощью команды `vi` следующим образом:

```
$ vi /tmp/passwd
```

Чтобы испытать все преимущества vim, убедитесь, что установлен расширенный пакет программ vim-enhanced (который по умолчанию входит в Ubuntu).

Во многих системах vi — это псевдоним команды vim. В Ubuntu обе команды запускают vim. Вы можете перепроверить это с помощью команды `alias`.

```
/bin/vi /tmp/text.txt
```

Рассмотрим еще несколько способов **запуска vi**:

```
$ vi +25 /tmp/inittab          Начать со строки номер 25
$ vi + /tmp/inittab            Начать редактирование с последней строки
$ vi +/tty /tmp/inittab        Начать с первой строки со слова tty
$ vi -r /tmp/inittab           Восстановить файл после сбоя в работе сессии
$ view /tmp/inittab            Редактировать файл в режиме только чтения
```

После окончания работы с vi нужно сохранить данные и выйти из программы, для чего существует несколько способов. Чтобы **сохранить файл до того, как выйти**, наберите `:w`. Если вы хотите **выйти и сохранить изменения**, наберите `ZZ` либо `:wq`. Чтобы **выйти без сохранения**, используйте команду `:q!`. Если вы обнаружите, что не можете изменять редактируемый файл, то, возможно, он открыт в режиме только для чтения. Если это так, то вы можете заставить программу записать данные, набрав `:w!`, или **сохранить содержимое файла под другим именем**. Например, наберите следующую команду, чтобы сохранить содержимое текущего файла в файл `myfile.txt`:

```
:w /tmp/myfile.txt
```

Редактор vi также позволяет одновременно **выбрать несколько файлов для редактирования**. Например, введите следующее:

```
$ cd /tmp
$ touch a.txt b.txt c.txt
$ vi a.txt b.txt c.txt
```

В этом примере vi откроет файл `a.txt` первым. Вы можете **перейти к следующему файлу**, набрав `:n`. Возможно, вам понадобится **сохранить изменения перед переходом к следующему файлу** (`:w`) или **сохранить изменения и перейти к следующему файлу** (`:wn`). Чтобы **отменить изменения и перейти к следующему файлу**, наберите `:n!`.

Скорее всего, будет легче открывать несколько файлов сразу, разделяя экран vi. Находясь в vi и открыв файл, можно горизонтально или вертикально **разделить экран несколько раз**:

```
:split /etc/motd.tail
:vsplit /etc/motd.tail
```

Используйте клавишу Tab, чтобы указать путь к файлам, как в обычном интерпретаторе команд bash. Чтобы **перемещаться между разделенными окнами**, нажмите сочетание Ctrl+W и клавишу W. Чтобы закрыть текущие окна, используйте обычную команду выхода из vi (`:q`).

Навигация

Первое, к чему следует привыкнуть, работая с vi, — это то, что нельзя сразу начать печатать. У vi есть множество режимов для выполнения различных типов задач.

Вы начинаете сессию vi в режиме Normal (Обычный), в то время как vi ждет команды для начала работы. Находясь в режиме Normal (Обычный), вы можете просматривать любое место в файле. Чтобы ввести или изменить текст, необходимо перейти в режим Insert (Вставка) или Replace (Замена).

Если в vi открыт файл с несколькими страницами текста, то для **навигации по файлу в режиме Normal (Обычный)** можно использовать клавиши и их сочетания, приведенные в табл. П1.1.

Таблица П1.1. Горячие клавиши для навигации

Клавиши	Описание	Клавиши	Описание
Page Down или Ctrl+F	Пролистать одну станицу вниз	Page Up или Ctrl+B	Пролистать одну станицу вверх
Ctrl+D	Пролистать полстраницы вниз	Ctrl+U	Пролистать полстраницы вверх
Shift+G	Перейти к последней строке файла	:L	Перейти к первой строке файла (используйте любую цифру для перехода на соответствующую строку)
Shift+H	Переместить курсор в верхнюю часть экрана	Shift+L	Переместить курсор в нижнюю часть экрана
Shift+M	Переместить курсор в среднюю часть экрана	Ctrl+L	Обновить изображение (если есть помехи)
Enter	Переместить курсор в начало следующей строки	–	Переместить курсор в начало предыдущей строки
Home или $	Переместить курсор в конец файла	End или ^ или 0	Переместить курсор в начало строки
(	Переместить курсор в начало предыдущего предложения	)	Переместить курсор в начало следующего предложения
{	Переместить курсор в начало предыдущего абзаца	}	Переместить курсор в начало следующего абзаца
W	Переместить курсор к следующему слову (пробел, новая строка или знак пунктуации)	Shift+W	Переместить курсор к следующему слову (пробел или новая строка)
B	Переместить курсор к предыдущему слову (пробел, новая строка или знак пунктуации)	Shift+B	Переместить курсор к предыдущему слову (пробел или новая строка)
E	Переместить курсор к концу следующего слова (пробел, новая строка или знак пунктуации)	Shift+E	Переместить курсор к концу следующего слова (пробел или новая строка)
← или Backspase	Переместить курсор на одну букву влево	→ или L	Переместить курсор на одну букву вправо
K или ↑	Переместить курсор на одну строку вверх	J или ↓	Переместить курсор на одну строку вниз
/string	Найти следующую комбинацию символов: string	?string	Найти предыдущую комбинацию символов строки
N	Искать далее	Shift+N	Искать в обратном направлении

Изменение и удаление текста

Чтобы начать изменять и добавлять текст с помощью vi, можно войти в режимы Insert (Вставка) или Replace (Замена), как показано в табл. П1.2. Когда вы переходите в эти режимы, набранные символы будут появляться в текстовом документе (в отличие от их интерпретации как команд).

Нажмите клавишу Esc, чтобы выйти обратно в режим Normal (Обычный) после того, как вы закончили вставлять или заменять текст.

Таблица П1.2. Клавиши для изменения текста

Клавиши	Описание	Клавиши	Описание
I	Набранный текст появляется перед текущим символом	Shift+I	Набранный тест появляется в начале текущей строки
A	Набранный текст появляется после текущего символа	Shift+A	Набранный тест появляется в конце текущей строки
O	Создать новую строку под текущей для начала ввода	Shift+O	Создать новую строку над текущей для начала ввода
S,	Стереть текущий символ и заменить новым текстом	Shift+S	Стереть текущую строку и заменить новым текстом
C?	Сменить ? на l, w, $ или c для изменения текущей буквы, слова, конца строки или строки	Shift+C	Стереть все от места установки курсора до конца строки и ввести новый текст
R	Заменить текущий символ следующим набранным символом	Shift+R	Заменить все по мере набора текста, начиная с текущего символа и далее вперед

Таблица П1.3 содержит клавиши для удаления или вставки текста.

Таблица П1.3. Клавиши для удаления и вставки текста

Клавиши	Описание	Клавиши	Описание
X	Удалить текст под курсором	Shift+X	Удалить текст слева от курсора
D?	Сменить ? на l, w, $ или d, чтобы вырезать текущий символ, слово или конец строки, начиная от курсора или целой строки	Shift+D	Вырезать все, начиная от места установки курсора до конца строки
Y?	Сменить ? на l, w или $, чтобы копировать (копировать в буфер) текущий символ, слово или конец строки, начиная от курсора	Shift+Y	Копировать текущую строку в буфер
P	Вставить вырезанный или скопированный в буфер текст после курсора	Shift+P	Вставить вырезанный или скопированный в буфер текст перед курсором

Вспомогательные команды

В табл. П1.4 показаны несколько вспомогательных, но важных для работы клавиш и их сочетаний.

Таблица П1.4. Вспомогательные клавиши

Клавиши	Описание
U	Отменить предыдущее изменение. Несколько нажатий U подряд отменят соответствующее количество действий
.	Повторить следующую команду. Таким образом, если вы удалили строку, заменили слово, изменили четыре буквы и т. д., та же самая команда будет выполнена с того места, где установлен курсор (команда сбрасывается при повторном входе в режим ввода)
Shift+J	Объединить текущую строку со следующей
Esc	Возвратиться в командный режим из режима ввода. Это одна из клавиш, используемых наиболее часто

Модификация команд с помощью чисел

Почти каждая описанная выше команда может быть подвержена модификации с помощью цифр. Другими словами, вместо удаления одного слова, замены буквы или изменения строки вы можете удалить шесть слов, заменить двенадцать букв и изменить девять строк. В табл. П1.5 показаны примеры.

Таблица П1.5. Модификация команд с помощью чисел

Клавиши	Описание
7CW	Удалить следующие семь слов и заменить набранным текстом
5, Shift+D	Вырезать следующие пять строк (включая текущую)
3P	Вставить удаленный до этого текст три раза после курсора
9DB	Вырезать девять слов перед курсором
10J	Опустить курсор вниз на десять строк
Y2)	Копировать (копировать в буфер) текст, начиная от курсора и до конца следующих двух предложений
5, Ctrl+F	Перейти вперед на пять страниц
6, Shift+J	Объединить следующие шесть строк

Из этих примеров видно, что большинство горячих клавиш vi, предназначенных для изменения текста, удаления или навигации, могут быть модифицированы с помощью чисел.

Ex-команды

Редактор vi был построен на основе редактора под названием Ex. Некоторые команды vi, описанные выше, начинаются с точки с запятой и известны как Ex-команды.

Для ввода таких команд начните работу в режиме Normal (Обычный) и наберите двоеточие (:). Это позволит перейти в режим командной строки. В этом режиме можно использовать клавишу Tab для завершения команды или имени файла, а также клавиши управления курсором для навигации по журналу команд, как в интерпретаторе команд bash. Нажимая в конце команды Enter, вы переходите в режим Normal (Обычный).

В табл. П1.6 показаны примеры Ex-команд.

Таблица П1.6. Примеры Ex-команд

Команда	Описание
:!bash	Перейти в интерпретатор команд bash. Когда необходимо, наберите exit для возврата в vi
:!date	Запустить date (или любую команду по вашему выбору). Нажмите Enter для возврата
:!!	Повторно запустить предыдущую команду
:20	Перейти к строке 20 в файле
:5,10w abc.txt	Вписать строки с пятой по десятую в файл abc.txt
:e abc.txt	Выйти из текущего файла и начать редактировать файл abc.txt
:.r def.txt	Поместить содержимое def.txt ниже текущей строки в текущем файле
:s/RH/RedHat	Изменить Red Hat при первом совпадении на RH в текущей строке
:s/RH/Red Hat/g	Изменить Red Hat при всех совпадениях на RH в текущей строке
:%s/RH/Red Hat/g	Изменить Red Hat при всех совпадениях на RH во всем файле
:g/Red Hat/p	Просмотреть каждую строку в файле, содержащую "Red Hat"
:g/gaim/s//pidgin/gp	Найти каждую реализацию gaim и изменить на pidgin

В командной строке ех вы также можете просмотреть и изменить настройки сессии vi, используя команду set. В табл. П1.7 показаны примеры.

Таблица П1.7. Команда set в ex-режиме

Команда	Описание
:set all	Вывести список всех настроек
:set	Показать только те настройки, которые были изменены с используемых по умолчанию на новые
:set number	Вывести номера строк слева от каждой строки (используйте set nonu для сброса)
:set ai	Установить автоструктурирование; таким образом, новая строка структурируется в соответствии с предыдущей
:set ic	Игнорировать регистр, чтобы поиск текста производился вне зависимости от регистра
:set list	Показывать $ на конце каждой строки и ^I для позиции табуляции
:set wm	Позволить vi добавлять разрывы между словами рядом с концом строки

Работа в графическом режиме

Редактор vim предоставляет интуитивный метод выбора текста, называемый графическим режимом. Чтобы начать работу в графическом режиме, поставьте курсор на первый символ нужного текста и нажмите клавишу V. Вы увидите, что вошли в графический режим, так как вверху появится текст:

```
-- VISUAL --
```

Теперь можно использовать любые клавиши управления курсором (стрелки, **Page Down**, **End** и т. д.) для перемещения курсора к концу текста, который вы хотите выбрать. Вы увидите, что, по мере того как страница и курсор двигаются, текст выделяется. Когда весь необходимый текст выделен, можете выполнять команды для работы с ним. Например, `d` удаляет текст, `c` позволяет изменять выбранный текст, `:w /tmp/test.txt` сохраняет выбранный текст в файл и т. д.

Приложение 2. Специальные символы и переменные интерпретатора команд

В Ubuntu интерпретатор команд bash используется по умолчанию. В гл. 3 рассказано, как работать с ним. В этом приложении приводятся ссылки на множество символов и переменных, которые имеют особое значение в интерпретаторе команд bash. Многие их этих элементов представлены в табл. П2.1 и П2.2.

Специальные символы интерпретатора команд

Вы можете использовать специальные символы из интерпретатора команд для сравнения нескольких файлов, сохранения сочетаний горячих клавиш или для выполнения других операций. В табл. П2.1 показаны некоторые полезные специальные символы интерпретатора команд.

Таблица П2.1. Специальные символы интерпретатора команд

Символы	Описание
*	Обозначает любой набор символов
?	Определяет любой символ
[...]	Обозначает любой символ, заключенный в скобки
' ... '	Указывает удалить специальное значение символа в скобках. Переменные не расширяются
" ... "	То же самое, что и одинарные кавычки, только символы перехода ($ ` и \) сохраняют специальное значение
\	Символ перехода для удаления специального значения последующего символа
~	Ссылка на папку $HOME
~+	Значение переменной PWD интерпретатора команд (рабочая директория)

Продолжение ⇨

Таблица П2.1 (продолжение)

Символы	Описание
~–	Ссылка на предыдущую рабочую папку
.	Ссылка на текущую рабочую папку
..	Ссылка на папку, расположенную над текущей. Символ может быть использован несколько раз подряд для указания папок, находящихся выше в дереве каталогов
$param	Используется для расширения параметра переменной интерпретатора команд
cmd1 \`cmd2\` или cmd1 $(cmd2)	cmd2 выполняется первым. Затем вызов cmd2 заменяется результатом выполнения cmd2 и выполняется cmd1
cmd1 >	Перенаправляет стандартный вывод команды
cmd1 <	Перенаправляет стандартный ввод команды
cmd1 >>	Прикрепляет стандартный вывод команды в файл, не удаляя его текущего содержимого
cmd1 | cmd2	Связывает вывод одной команды и ввод другой
cmd &	Запускает команду в фоновом режиме
cmd1 && cmd2	Запускает первую команду. Если получено нулевое значение, запускает вторую команду
cmd1 || cmd2	Запускает первую команду. Если не получено нулевое значение, запускает вторую команду
cmd1 ; cmd2	Запускает первую команду и, когда ее выполнение завершается, запускает вторую

Переменные интерпретатора команд

Вы можете создать параметр (переменную), поставив $ перед цепочкой символов (как в $HOME). Переменные среды интерпретатора команд могут хранить информацию, используемую самим интерпретатором, а также командами, которые вы запускаете из него. Не все переменные среды будут наполнены по умолчанию. Некоторые переменные можно изменять (как принтер по умолчанию $PRINTER или командную строку в $PS1). Другие управляются интерпретатором команд (например, $OLDPWD). В табл. П2.2 приводится список полезных переменных интерпретатора команд.

Таблица П2.2. Переменные интерпретатора команд

Переменная	Описание
BASH	Показывает путь к команде bash (/bin/bash)
BASH_COMMAND	Команда, выполняемая в текущий момент
BASH_VERSION	Версия bash
COLORS	Путь к файлам конфигурации цветов ls
COLUMNS	Ширина строки терминала (в символах)

Переменная	Описание
DISPLAY	Обозначает X-экран, в котором будут отображаться команды, запущенные из текущего интерпретатора команд (например, :0.0)
EUID	Идентификационный номер текущего пользователя. Основывается на записи в /etc/passwd для вошедшего в систему пользователя
FCEDIT	Определяет текстовый редактор, использующийся командой fc для редактирования команд history. Редактор vi используется по умолчанию
GROUPS	Выводит список групп, в которые входит текущий пользователь
HISTCMD	Показывает текущий номер журнальной записи текущей команды
HISTFILE	Отображает местонахождение файла журнала (обычно находится в папке $HOME/.bash_history)
HISTFILESIZE	Общее количество записей в журнале для хранения (1000 по умолчанию). После достижения этого числа старые команды не учитываются
HISTCMD	Номер текущей команды в файле журнала
HOME	Местонахождение домашней директории текущего пользователя. Команда cd без параметров возвращает в домашний каталог
HOSTNAME	Имя хост-узла текущей машины
HOSTTYPE	Содержит архитектуру компьютера, на которой работает Linux (i386, i486, i586, i686, x86_64, ppc или ppc64)
LESSOPEN	Приписана команде, конвертирующей содержание файла, отличное от чистого текста (рисунки, RPM, ZIP-файлы и т. д.), для работы с командой less
LINES	Устанавливает количество строк в текущем терминале
LOGNAME	Содержит имя текущего пользователя
LS_COLORS	Приписывает определенные цвета расширениям файлов; в результате команда ls их отображает, когда сталкивается с ними
MACHTYPE	Отображает информацию об архитектуре машины, компании и операционной системе (например, i686-redhat-linux-gnu)
MAIL	Обозначает местонахождение файла почтового ящика (обычно это имя пользователя в папке/var/spool/mail)
MAILCHECK	Проверяет почту с указанными интервалами (60 по умолчанию)
OLDPWD	Папка, которая являлась рабочей до смены на текущую директорию
OSTYPE	Имя, обозначающее текущую операционную систему (как linux или linux-gnu)
PATH	Список папок, разделенных двоеточиями, используемый для нахождения источника набираемых команд (/bin, /usr/bin и $HOME/bin обычно входят в PATH)
PPID	Идентификационный номер процесса команды, запустившей этот интерпретатор команд
PRINTER	Установить принтер, применяемый по умолчанию, который используется такими командами, как lpr и lpq

Продолжение ⇗

Таблица П2.2 (продолжение)

Переменная	**Описание**
PROMPT_COMMAND	Установить имя команды для запуска перед появлением приглашения командного процессора. Например, PROMPT_COMMAND=ls выводит список команд в текущей папке перед отображением командной строки
PS1	Устанавливает приглашение командного процессора. Оно может включать в себя даты, время, имена пользователей, имя хост-узла и т. д. Дополнительные приглашения командного процессора могут быть установлены с помощью PS2, PS3 и т. д.
PWD	Эта папка является текущей
RANDOM	Данная переменная генерирует случайное число от 0 до 32 767
SECONDS	Количество секунд с начала работы интерпретатора команд
SHELL	Содержит полный путь текущего интерпретатора команд
SHELLOPTS	Вывести список включенных параметров интерпретатора команд

Приложение 3. Получение информации с помощью файловой системы /proc

Файловая система /proc изначально предназначалась для хранения информации, используемой запущенными процессами. В конце концов она стала главным местом хранения информации, используемой ядром Linux. Несмотря на появление /sys для вывода более упорядоченной информации о состоянии ядра, многие утилиты Linux все еще собирают и представляют информацию о работающей системе с помощью /proc.

Если вы не любите выполнять лишние действия, то можете не использовать улилиты, которые обращаются к файлам /proc и иногда проводят записи в /proc. Просматривая /proc, можно узнать текущее состояние процессов, устройств, подсистем ядра и других атрибутов Linux.

Просмотр информации

Просмотр информации в файлах папки /proc можно осуществить с помощью простой команды cat. В папке /proc существуют отдельные директории для каждого работающего процесса (названные в соответствии с их идентификаторами), которые содержат информацию о процессе. Кроме того, в ней хранятся файлы, содержащие данные о некоторых других свойствах системы, например о процессоре, состоянии оперативной памяти, версиях программ, разделов дисков и т. д.

Следующие примеры показывают, какую информацию можно получить из папки /proc в системе Linux:

```
$ cat /proc/cmdline         Показывает параметры, переданные ядру при загрузке
root=UUID=db2dac48-a62e-4dbe-9529-e88a57b15bac ro quiet splash
$ cat /proc/cpuinfo         Показывает информацию о процессоре
Processor : 0
vendor_id : GenuineIntel cpu family : 6
model           : 8
model name      : Pentium III (Coppermine)
stepping        : 3
cpu MHz         : 648.045
cache size      : 256 KB
```

В этом примере скорость может быть намного ниже, чем реальная, если запущен оптимизатор процессора, например cpuspeed.

```
$ cat /proc/devices           Отображает существующие устройства посимвольного
                              ввода-вывода и блочные устройства
Character devices:
   1 mem
   4 /dev/vc/0
   4 tty
   4 ttys
   5 /dev/tty
   ...
Block devices:
   1 ramdisk
   8 sd
   9 md
$ cat /proc/diskstats         Отображает диски, разделы и статистику по ним
   1      0 ram0 0 0 0 0 0 0 0 0 0 0 0
   1      1 ram1 0 0 0 0 0 0 0 0 0 0 0
   ...
   8      0 sda 2228445 1032474 68692149 21672710 1098740 4003143
47790770 101074392 0 15385988 122799055
   8      1 sda1 330077 13060510 188002 8443280
   8      1 sda2 1491 1759 50 162
   ...
   7      0 loop 0 0 0 0 0 0 0 0 0 0 0
```

В только что показанном выводе команды diskstats можно увидеть псевдодиск (ram0, ram1 и т. д.) и циклические устройства (loop0, loop1 и т. д.). Что касается разделов жестких дисков, то пример отображает статистику по целому жесткому диску (sda) и каждому разделу (sda1, sda2 и т. д.).

Поле 11 информации о жестком диске отображает (слева направо): общее количество чтений; количество объединенных чтений; прочитанных секторов; миллисекунд, затрачиваемых каждым чтением; завершенных записей; объединенных записей; секторов, в которые производилась запись; миллисекунд, затрачиваемых на запись; текущих запросов ввода-вывода; миллисекунд, затраченных на ввод-вывод; среднее количество миллисекунд, затраченных на ввод-вывод. Поля для определенных разделов показывают (слева направо): количество заданных чтений; прочитанных секторов; заданных записей; секторов, в которых произведена запись.

```
$ cat /proc/filesystems       Вывести список поддерживаемых ядром файловых систем
nodev     sysfs               nodev означает, что данный тип не используется
                              ни одним устройством
nodev     rootfs

          ext3                ext3   используется на смонтированном
                                     блочном устройстве
          iso9660             iso9660 используется на смонтированном
                                     блочном устройстве
```

```
$ cat /proc/interrupts      Просмотр присвоенных IRQ каналов
         CPU0
   0:    198380901   XT-PIC-XT   timer
   1:    28189       XT-PIC-XT   i8042
   2:    0                 XT-PIC-XT   cascade
   6:    3770197           XT-PIC-XT   Ensoniq AudioPCI
   7:    660               XT-PIC-XT   parport0
   ...
$ cat /proc/iomem           Показать адреса физической памяти
00000000-0009fbff : System RAM
00000000-00000000 : Crash kernel
0009fc00-0009ffff : reserved
000a0000-000bffff : Video RAM area
000c0000-000c7fff : Video ROM
000c8000-000c8fff : Adapter ROM
000f0000-000fffff : System ROM
00100000-0febffff : System RAM
$ cat /proc/ioports         Показать адреса виртуальной памяти
0000-001f : dma1
0020-0021 : pic1
0040-0043 : timer0
0050-0053 : timer1
0060-006f : keyboard
0070-0077 : rtc
0080-008f : dma page reg
00a0-00a1 : pic2
00c0-00df : dma2
00f0-00ff : fpu
...
$ cat /proc/loadavg         Показать 1-, 5- и 15-минутные средние загрузки,
1.77 0.56 0.19 2/247 1869   запущенные процессы/общий и наивысший PID
$ cat /proc/meminfo         Показать доступную оперативную память
                            и память подкачки
MemTotal: 482992 kB
MemFree:   25616 kB
Buffers:   12204 kB
Cached:          64132 kB
SwapCached:     117472 kB
Active:         321344 kB
Inactive:  93168 kB
HighTotal:             0 kB
HighFree:        0 kB
LowTotal: 482992 kB
   ...
$ cat /proc/misc            Показать имя/младший номер устройств,
229 fuse                    зарегистрированных главным устройством misc (10)
 63 device-mapper
175 agpgart
144 nvram
```

$ cat /proc/modules *Показать загруженные модули, размер памяти, загруженные копии программы, состояние загруженных зависимостей и память ядра*

```
nls_utf8 6209 1 - Live 0xd0c59000
cifs 213301 0 - Live 0xd0e3b000
nfs 226861 0 - Live 0xd0e02000
nfsd 208689 17 - Live 0xd0d8a000
exportfs 9537 1 nfsd, Live 0xd0cfb000
lockd 62409 3 nfs,nfsd, Live 0xd0d45000
nfs_acl 7617 2 nfs,nfsd, Live 0xd0c56000
fuse 45909 2 - Live 0xd0d24000
vfat 16193 0 - Live 0xd0cf6000
   ...
```

$ cat /proc/mounts *Показать информацию о смонтированных локальных/удаленных файловых системах*

```
rootfs / rootfs rw 0 0
none /sys sysfs rw,nosuid,nodev,noexec 0 0 none /proc proc rw,nosuid,nodev,noexec
0 0 udev /dev tmpfs rw 0 0
/dev/disk/by-uuid/db2dac48-a62e-4dbe-9529-e88a57b15bac / ext3 rw,data=
ordered 0 0
/dev/disk/by-uuid/db2dac48-a62e-4dbe-9529-e88a57b15bac /dev/.static/dev ext3
rw,data=ordered 0 0
tmpfs /var/run tmpfs rw,nosuid,nodev,noexec  0 0
tmpfs /var/lock tmpfs rw,nosuid,nodev,noexec 0 0
tmpfs /lib/modules/2.6.20-16-generic/volatile tmpfs rw 0 0
tmpfs /dev/shm tmpfs rw 0 0
devpts /dev/pts devpts rw 0 0
usbfs /dev/bus/usb/.usbfs usbfs rw 0 0
udev /proc/bus/usb tmpfs rw 0 0
usbfs /proc/bus/usb/.usbfs usbfs rw 0 0
fusectl /sys/fs/fuse/connections fusectl rw 0 0 tmpfs /var/run tmpfs
rw,nosuid,nodev,noexec 0 0 tmpfs /var/lock tmpfs rw,nosuid,nodev,noexec 0 0
/dev/disk/by-uuid/4f419cb8-a920-4b6e-a8fd-b3946f9bf644 /boot ext3
rw,data=ordered 0 0
/dev/disk/by-uuid/91ae7a92-ca5c-4ef0-9729-ba0cdcf2a07f /home2 ext3
rw,data=ordered 0 0
nfsd /proc/fs/nfsd nfsd rw 0 0
rpc_pipefs /var/lib/nfs/rpc_pipefs rpc_pipefs rw 0 0
binfmt_misc /proc/sys/fs/binfmt_misc binfmt_misc rw 0 0
```

$ cat /proc/partitions *Показать смонтированные локальные разделы*

```
major    minor #blocks          name
8           0             40031712       sda
8           1             200781         sda1
8           2             10241437       sda2
8           3             6160927        sda3
     ...
```

$ cat /proc/mdstat *Если используется программное обеспечение RAID, показать статус RAID*

```
Personalities : [raid1]
read_ahead 1024 sectors
Event: 1
md0 : active raid1 sdb1[1] sda2[0]
     69738048 blocks [2/2] [UU]
unused devices: <none>
```

Файл /proc/mdstat содержит детальную информацию о положении программных устройств RAID, если вы их устанавливали. В этом примере md0 — это RAID1 (зеркало), составленный из разделов /dev/sdb1 и /dev/sda1. В этой строке присутствует буква U для каждого рабочего RAID-устройства. Если вы потеряете диск, вывод будет выглядеть как [U_].

```
$ cat /proc/stat        Отобразить статистику ядра с момента запуска системы
cpu  1559592 1488475 710279 218584583 1446866 5486 16708
cpu0 1559592 1488475 710279 218584583 1446866 5486 16708
intr 215956694 200097282 28242 0 1 3 0 3770197 660 1 1 0 3753340 ...
ctxt 281917622
btime 1181950070 processes 519308 procs_running 1 procs_blocked 0
```

Файл /proc/stat содержит статистику активности процессов и системного процессора. Строка cpu отображает общее количество всех процессоров, в то время как отдельные строки для каждого процессора (cpu0, cpu1 и т. д.) показывают статистику по каждому процессору на компьютере. Есть семь полей (слева направо) информации о процессоре: количество обычных процессов, выполняемых в режиме пользователя; niced-процессы, запущенные в режиме пользователя; процессы ядра; свободные процессы; iowait-процессы (ожидающие ввода-вывода для завершения); прерванные службы (IRQ); программы обслуживания IRQ.

```
$ cat /proc/swaps       Отобразить информацию о размере файла подкачки
Filename Type  Size  Used  Priority
/dev/sda2 partition 1020088  201124 -1
$ cat /proc/uptime      Секунды с момента загрузки системы/общее
                        количество секунд простоя
2300251.03 2261855.31
$ cat /proc/version     Вывести версию ядра и соответствующего компилятора
Linux version 2.6.20-16-generic (root@terranova) (gcc version 4.1.2 (Ubuntu
4.1.2-0ubuntu4)) #2 SMP Fri Aug 31 00:55:27 UTC 2007
```

Изменение информации

В некоторых версиях Linux определенные значения в папке /proc/sys могут быть изменены в процессе работы системы. Команда sysctl — это наиболее удачный метод изменения информации /proc/sys. Для окончательного изменения этих настроек нужно добавить записи в файл /etc/sysctl.conf. Вот несколько примеров команды sysctl:

```
$ sudo sysctl -A | less        Отобразить все динамические параметры ядра
$ sudo sysctl -w net.ipv4.ip_forward=1   Включить перенаправление пакетов IPV4
```

Обращайтесь к руководствам sysctl и sysctl.conf, а также к гл. 10 для получения более подробной информации.

Алфавитный указатель

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

W

X

А

Б

М

О

П

Р

С

Т

У

Ф

Х

Ц

Э

Я

Негус К., Каэн Ф.

Ubuntu и Debian Linux для продвинутых: более 1000 незаменимых команд

Перевели с английского Ю. Зверев, А. Одноочко

Заведующий редакцией	*А. Громаковский*
Ведущий редактор	*Н. Гринчик*
Литературные редакторы	*М. Андреева, Д. Романов*
Художник	*К. Радзевич*
Корректоры	*В. Субот, Ю. Цеханович*
Верстка	*Г. Блинов*

Подписано в печать 09.08.10. Формат 70×100/16. Усл. п. л. 28,38. Доп. тираж 1000. Заказ 23428.

ООО «Лидер», 194044, Санкт-Петербург, Б. Сампсониевский пр., 29а.

Налоговая льгота — общероссийский классификатор продукции ОК 005-93, том 2; 95 3005 — литература учебная.

Отпечатано по технологии CtP в ОАО «Печатный двор» им. А. М. Горького.
197110, Санкт-Петербург, Чкаловский пр., 15.


ИЗДАТЕЛЬСКИЙ ДОМ
ПИТЕР®
WWW.PITER.COM



УВАЖАЕМЫЕ ГОСПОДА!
КНИГИ ИЗДАТЕЛЬСКОГО ДОМА «ПИТЕР»
ВЫ МОЖЕТЕ ПРИОБРЕСТИ
ОПТОМ И В РОЗНИЦУ У НАШИХ
РЕГИОНАЛЬНЫХ ПАРТНЕРОВ.

ДАЛЬНИЙ ВОСТОК

Владивосток
«Приморский торговый дом книги»
тел./факс: (4232) 23-82-12
e-mail: bookbase@mail.primorye.ru

Хабаровск, «Деловая книга», ул. Путевая, д. 1а
тел.: (4212) 36-06-65, 33-95-31
e-mail: dkniga@mail.kht.ru

Хабаровск, «Книжный мир»
тел.: (4212) 32-85-51, факс: (4212) 32-82-50
e-mail: postmaster@worldbooks.kht.ru

Хабаровск, «Мирс»
тел.: (4212) 39-49-60
e-mail: zakaz@booksmirs.ru

ЕВРОПЕЙСКИЕ РЕГИОНЫ РОССИИ

Архангельск, «Дом книги», пл. Ленина, д. 3
тел.: (8182) 65-41-34, 65-38-79
e-mail: marketing@avfkniga.ru

Воронеж, «Амиталь», пл. Ленина, д. 4
тел.: (4732) 26-77-77
http://www.amital.ru

Калининград, «Вестер»,
сеть магазинов «Книги и книжечки»
тел./факс: (4012) 21-56-28, 6 5-65-68
e-mail: nshibkova@vester.ru
http://www.vester.ru

Самара, «Чакона», ТЦ «Фрегат»
Московское шоссе, д.15
тел.: (846) 331-22-33
e-mail: chaconne@chaccone.ru

Саратов, «Читающий Саратов»
пр. Революции, д. 58
тел.: (4732) 51-28-93, 47-00-81
e-mail: manager@kmsvrn.ru

СЕВЕРНЫЙ КАВКАЗ

Ессентуки, «Россы», ул. Октябрьская, 424
тел./факс: (87934) 6-93-09
e-mail: rossy@kmw.ru

СИБИРЬ

Иркутск, «ПродаЛитЪ»
тел.: (3952) 20-09-17, 24-17-77
e-mail: prodalit@irk.ru
http://www.prodalit.irk.ru

Иркутск, «Светлана»
тел./факс: (3952) 25-25-90
e-mail: kkcbooks@bk.ru
http://www.kkcbooks.ru

Красноярск, «Книжный мир»
пр. Мира, д. 86
тел./факс: (3912) 27-39-71
e-mail: book-world@public.krasnet.ru

Новосибирск, «Топ-книга»
тел.: (383) 336-10-26
факс: (383) 336-10-27
e-mail: office@top-kniga.ru
http://www.top-kniga.ru

ТАТАРСТАН

Казань, «Таис»,
сеть магазинов «Дом книги»
тел.: (843) 272-34-55
e-mail: tais@bancorp.ru

УРАЛ

Екатеринбург, ООО «Дом книги»
ул. Антона Валека, д. 12
тел./факс: (343) 358-18-98, 358-14-84
e-mail: domknigi@k66.ru

Екатеринбург, ТЦ «Люмна»
ул. Студенческая, д. 1в
тел./факс: (343) 228-10-70
e-mail: igm@lumna.ru
http://www.lumna.ru

Челябинск, ООО «ИнтерСервис ЛТД»
ул. Артиллерийская, д. 124
тел.: (351) 247-74-03, 247-74-09,
247-74-16
e-mail: zakup@intser.ru
http://www.fkniga.ru, www.intser.ru

Узнайте огромное количество самых нужных команд для Ubuntu Linux!

Это руководство научит вас использовать Ubuntu Linux так, как это делают настоящие профессионалы, то есть с помощью командной строки. Вы сможете применять более 1000 команд и получите все необходимое программное обеспечение — начиная с системных утилит, отслеживающих работу вашего ПК и его безопасность, и заканчивая программами для работы в сети и разграничения доступа. Книга дает незаменимые знания и навыки для использования и администрирования настольных ПК и серверов, работающих под управлением Ubuntu, Debian, KNOPPIX и других дистрибутивов Linux.

Кристофер Негус — автор множества бестселлеров, среди которых книги по дистрибутивам Fedora и Red Hat Linux. Является членом Мэдисонского сообщества пользователей Linux (*Madison Linux User Group*). Прежде чем заняться писательской деятельностью, он работал над созданием Unix в компании *AT&T* — родоначальнике этой операционной системы. Кроме того, он трудился в компании *Novell* над разработкой их собственной ОС Caldera.

Франсуа Каэн — владелец компании *Turbosphere LLC*, где специализируется на поддержке и управлении инфраструктурных приложений. Является защитником движения Open Source, сертифицированным инженером Rad Hat Linux (RHCE) и президентом Tacoma Linux User Group.

Тема: Операционные системы/Linux

Уровень пользователя: опытный

Заказ книг:

197198, Санкт-Петербург, а/я 127
тел.: (812) 703-73-74, postbook@piter.com

61093, Харьков-93, а/я 9130
тел.: (057) 758-41-45, 751-10-02,
piter@kharkov.piter.com





www.piter.com — вся информация о книгах и веб-мага